

■ PROJECTS ■ THEORY ■ APPLICATIONS ■ CIRCUITS ■ TECHNOLOGY

NUTS AND VOLTS

NUTS AND VOLTS

www.nutsvolts.com
October 2013

EVERYTHING FOR ELECTRONICS

Pick An App, Any App!

Choose A Sensor For All Kinds Of *Wireless*

Detection And Alarm Applications

- ◆ IR
- ◆ Tilt
- ◆ Light
- ◆ Sound
- ◆ Water
- ◆ Humidity
- ◆ Vibration
- ◆ Proximity
- ◆ Temperature



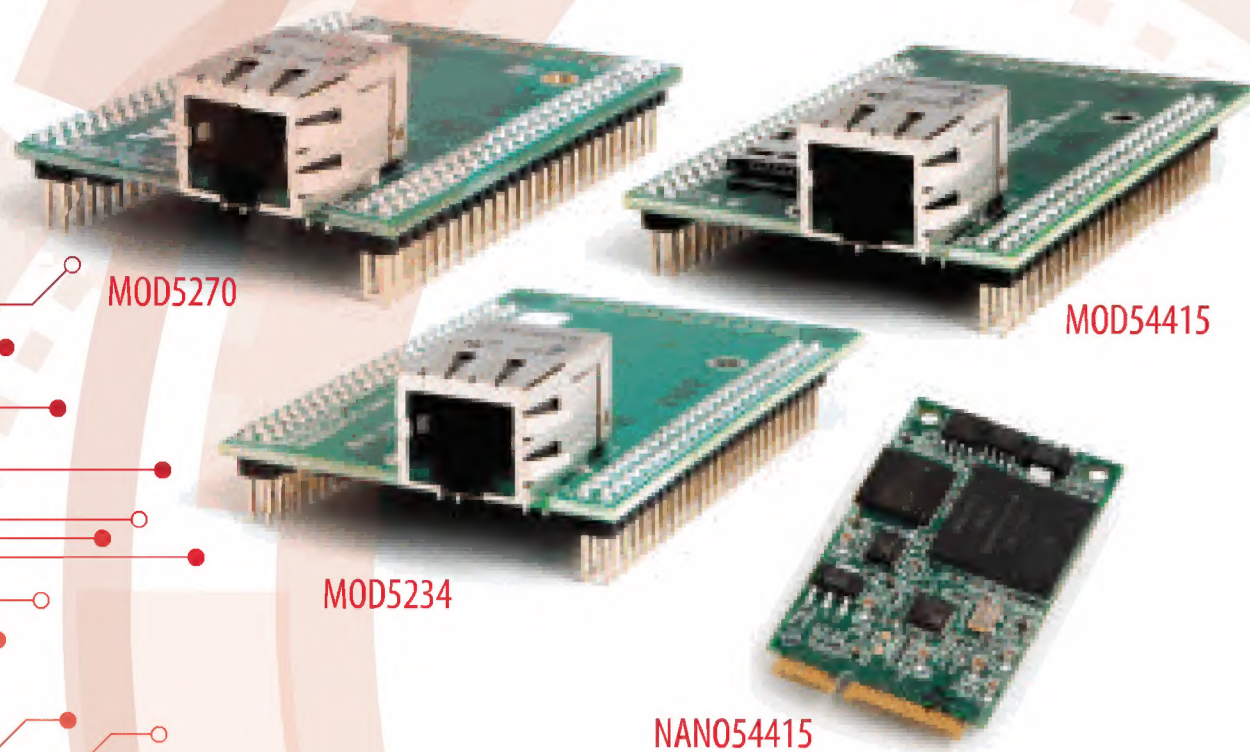
U.S. \$6.50 CANADA \$7.50



0 71486 02421 7

10>

Ethernet Core Modules with High-Performance Connectivity Options

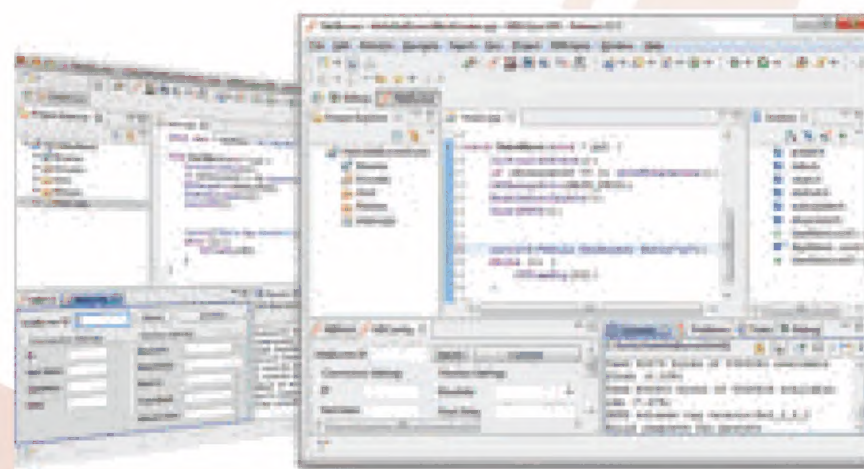


- **MOD5270**
147.5 MHz processor with 512KB Flash & 8MB RAM · 47 GPIO · 3 UARTs · I²C · SPI
- **MOD5234**
147.5 MHz processor with 2MB flash & 8MB RAM · 49 GPIO · 3 UARTs · I²C · SPI · CAN · eTPU (for I/O handling, serial communications, motor/timing/engine control applications)
- **MOD54415**
250 MHz processor with 32MB flash & 64MB RAM · 44 GPIO · 8 UARTs · 5 I²C · 3 SPI · 2 CAN · SSI · 8 ADC · 2 DAC · 8 PWM · 1-Wire® interface
- **NANO54415**
250 MHz processor with 8MB flash & 64MB RAM · 30 GPIO · 8 UARTs · 4 I²C · 3 SPI · 2 CAN · SSI · 6 ADC · 2 DAC · 8 PWM · 1-Wire® interface

Add Ethernet connectivity to an existing product, or use it as your product's core processor



The goal: Control, configure, or monitor a device using Ethernet



The method: Create and deploy applications from your Mac or Windows PC. Get hands-on familiarity with the NetBurner platform by studying, building, and modifying source code examples.



The result: Access device from the Internet or a local area network (LAN)

The NetBurner Ethernet Core Module is a device containing everything needed for design engineers to add network control and to monitor a company's communications assets. For a very low price point, this module solves the problem of network-enabling devices with 10/100 Ethernet, including those requiring digital, analog and serial control.

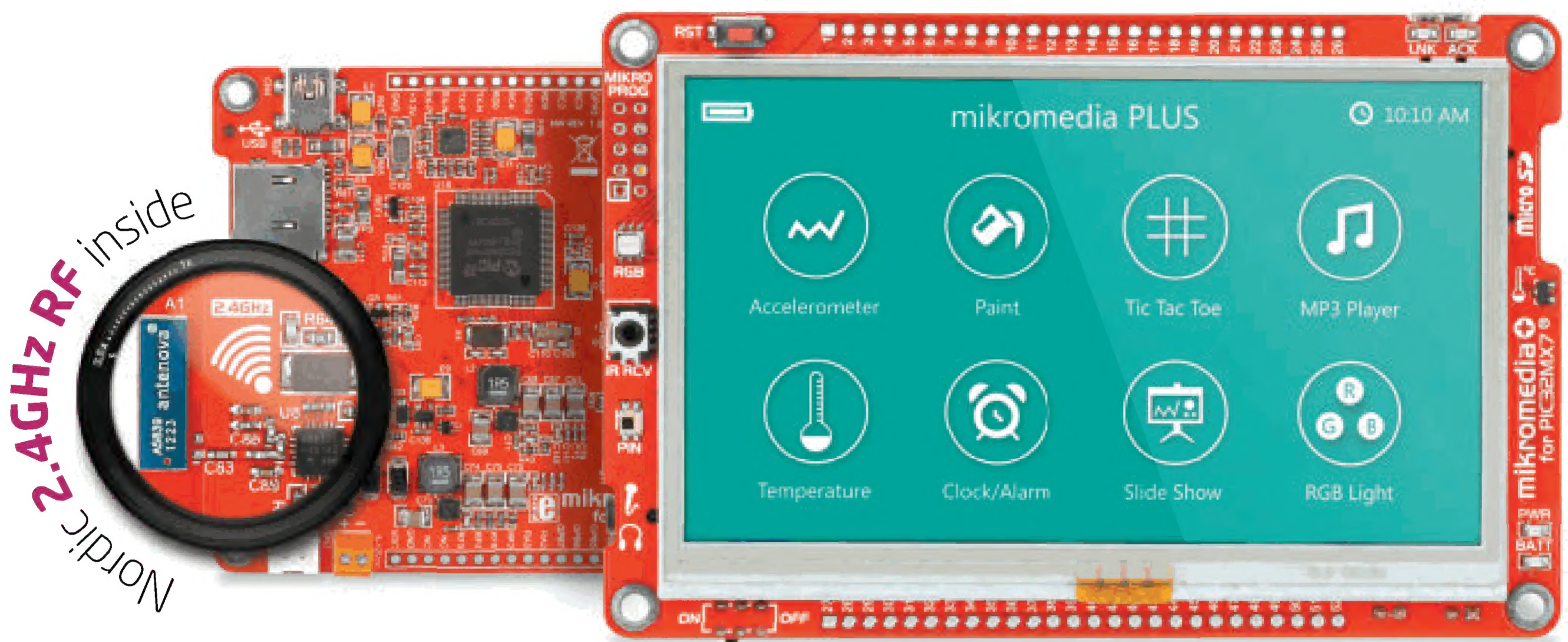
MOD5270-100IR.....\$69 (qty. 100)	NNDK-MOD5270LC-KIT\$99
MOD5234-100IR.....\$99 (qty. 100)	NNDK-MOD5234LC-KIT\$249
MOD54415-100IR.....\$89 (qty. 100)	NNDK-MOD54415LC-KIT\$129
NANO54415-200IR...\$69 (qty. 100)	NNDK-NANO54415-KIT.....\$99

NetBurner Development Kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications. The kits include all the hardware and software you need to build your embedded application.

➤ **For additional information please visit**
<http://www.netburner.com/kits>

32-bits and **pieces**

...but who's counting



price: \$199⁰⁰

mikromedia⁺ for PIC32MX7[®]

Fiery red color is there for a reason. This piece of tech will make your geek adrenalin level rise up! With **32 different modules**, you'll be hardly missing anything. There are tons of stuff on-board: 4.3" touch screen, RF, accelerometer, Ethernet, sensors, mp3 codec, battery charger - you name it and it's there! All is driven by the mighty **PIC32MX795F512L microcontroller**. So, tell your wife to make sandwiches and coffee, 'cause you'll be up playing with this board all night long.

October 2013

28 Build the Wireless Super Detector!

This project started out with a singular focus and grew into a multi-purpose circuit with all kinds of sensors for detection and alarm applications.

■ By Ron Newton

36 Build the LED Matrix Game Player

With a little imagination and a whole lot of time on your hands, you can create simple — yet amazing — action games you can play anywhere!

■ By Matthew Bates



44 Get Amp'ed Up Over the Theory Behind Building a Headset or Power Amplifier

The last two issues have featured projects to construct amplifiers with different purposes. Read this to understand what makes them both tick.

■ By Ronald Anderson

48 MakerPlot — The DIY Software Kit

So, you've designed a microcontroller project. How do you get data out of it or do any debugging? Well, we've got the answer for you.

■ By John Gavlik and Martin Hebel

Departments

06 DEVELOPING PERSPECTIVES
Getting Out of Your Comfort Zone

07 READER FEEDBACK

26 NEW PRODUCTS

27 SHOWCASE

68 NV WEBSTORE

71 ELECTRO-NET

77 CLASSIFIEDS

78 TECH FORUM

81 AD INDEX

Columns

08 TechKnowledgey 2013

Events, Advances, and News

Brain sparks, USB speeding up, a free 3D printer (maybe), and digitizing the throne are just a few of the hot topics discussed this time around.

12 PICAXE Primer

Sharpening Your Tools of Creativity

Getting Started With Python Programming. We're going to begin our Raspberry Pi programming experiments by implementing a simple "Hello World!" program using Python, and then move on to the basics of controlling the RPi's GPIO pins with a Python program.

20 Q & A

Reader Questions Answered Here

A Tesla coil project, an infrared pulse amplifier, a mini LTspice tutorial, a backup camera mod, some newbie questions, and MailBag round out this month's coverage.

53 The Design Cycle

Advanced Techniques for Design Engineers

Programming the Brains Behind CNC Milling Tools and Motors.

Whether you're planning a custom CNC application or an embedded monitor and control project, these gizmos will help things add up in your automation.

61 Smiley's Workshop

Programming • Hardware • Projects

Arduino Handheld Prototyper — Part 4.

See how to get user feedback from the keys and display useful info on the LCD from the fresh air controller.

72 Open Communication

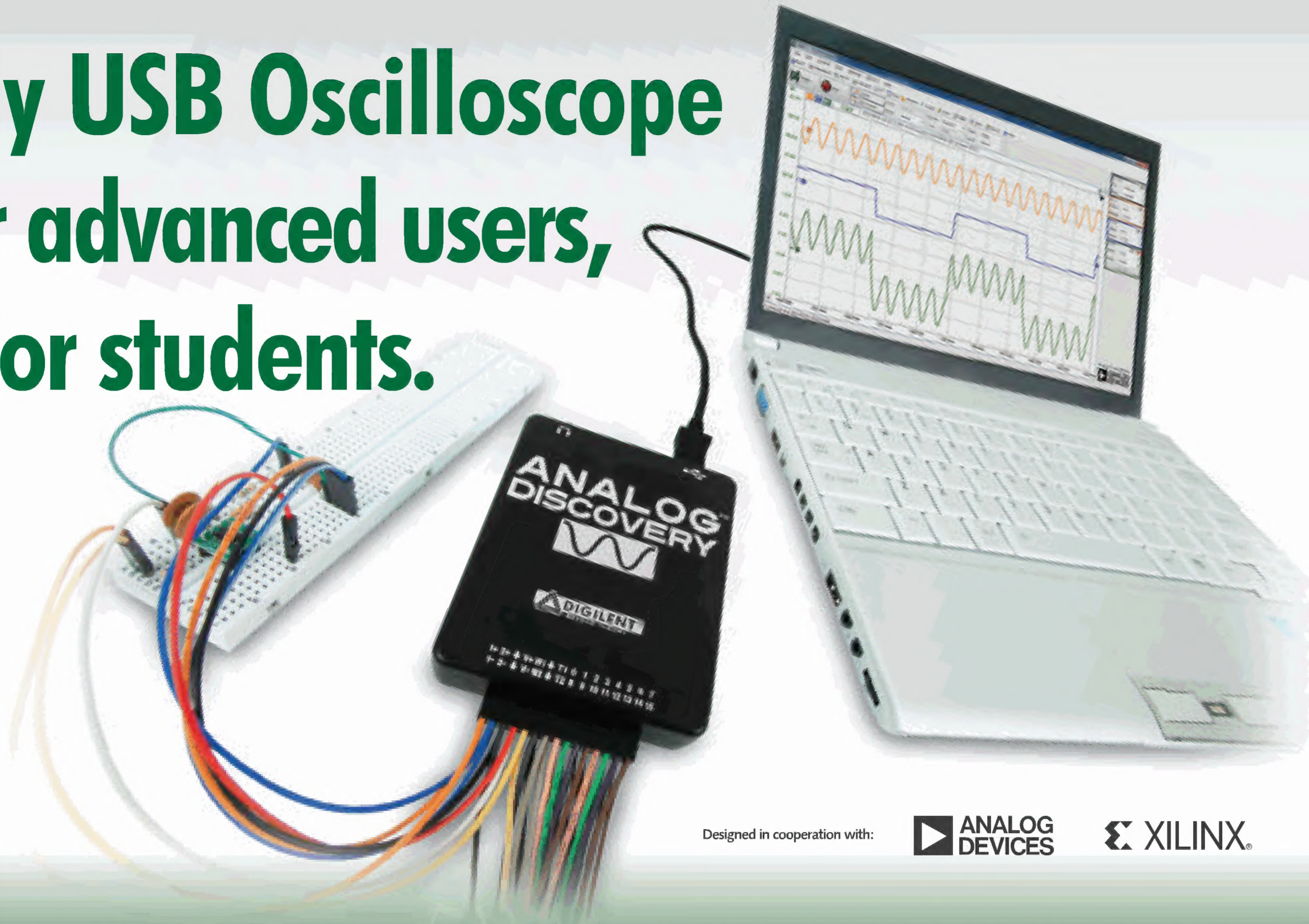
The Latest in Networking and Wireless Technologies

10 Things You Need to Know About Wi-Fi.

Discover facts about this familiar wireless technology that you may not already know.

The only USB Oscilloscope built for advanced users, priced for students.

\$99.00
(U.S. student price)



Designed in cooperation with:



Turn any PC into a powerful electrical engineering workstation! The USB-powered Analog Discovery lets you measure, visualize, analyze, record and control mixed signal circuits of all kinds. It's small enough to fit in your pocket, but powerful enough to replace a stack of lab equipment. Driven by the free WaveForms™ software, the Analog Discovery lets you build and test analog and digital circuits in virtually any environment, in or out of the lab.

- 2-Channel Oscilloscope
- 2-Channel Waveform Generator
- 16-Channel Logic Analyzer
- 16-Channel Digital Pattern Generator
- $\pm 5\text{VDC}$ Power Supplies
- Spectrum Analyzer
- Network Analyzer
- Voltmeter
- Digital I/O

Now supported by MATLAB / MATLAB student edition

Also Available:



Analog Parts Kit (APK)

- 20+ ICs from Analog Devices
- 200+ discrete components
- Solderless breadboard
- Perfect for Circuits 1&2 classes

\$69.99

Discovery BNC Adapter

- Allows the use of standard BNC-terminated probes
- Selectable AC & DC coupling

\$19.99



www.digilentinc.com/AnalogDiscovery

**ANALOG
DISCOVERY**
USB OSCILLOSCOPE+



by Bryan Bergeron, Editor

DEVELOPING PERSPECTIVES

Getting Out of Your Comfort Zone

I consider myself in decent physical shape — I bike daily and have a regular lifting routine. Whenever I visit my younger sister on the west coast, she pulls out some DVD on advanced yoga or pilates or Bulgarian bag training, and puts me to shame.

The trouble is — without external nudging — I tend to stay in my comfort zone at the expense of learning something new. With neuromuscular training, if you don't mix it up, you won't grow.

It's the same with technology. After I've spent months or years with a given microprocessor, I can usually do what needs to be done — even if it requires a few tricks here and there. Why invest the energy learning something new? In my case, it's simple. I need to keep up with the latest wave of microcontrollers, sensors, and other devices to do my job as editor.

It turns out that getting out of my comfort zone inevitably introduces me to techniques that I can bring back to my favorite microcontroller or analog chip set.

What about you? Lacking external pressure to leave your comfort zone puts the motivation completely in

your own hands. If you're proficient at cranking out circuits with discrete logic chips, why bother learning to program the PIC or a Raspberry Pi? Conversely, if you're a wiz at the Arduino, why bother with stand-alone A/D converter chips and high performance analog chips?

Well, I'm confident that you'll likely learn something worthwhile as you grope around uncharted technology landscapes.

Continuing with the example of an onboard A/D converter in microcontrollers, let's say you want to create an effects filter for a microphone or instrument pick-up. You could simply feed the audio signal to the A/D port of an Arduino. Once you've captured the signal, you could devise any number of digital filters to enhance or distort the signal before sending it out to an audio amp.

Of course, your filter design would be limited by the processing power and memory of the microcontroller. You might waste much of your microcontroller's filter capabilities by creating the equivalent of a low-pass filter — something easily and inexpensively done in the analog world with a capacitor and resistor.

Or, if you're after a distortion sound effect, you could do most of the work with a diode and resistor on the input of the A/D converter.

There's also the issue of noise. Laying out a microcontroller board doesn't take much forethought if you're making a blinking LED controller. There's input and output and power. Most often, experimenters with limited analog experience pay little attention to how the signals get where they're supposed to go.

However, if your expertise includes analog audio circuits, you know that signal routing is critical to minimize noise pickup — whether from the 60-cycle power mains, a poorly regulated power supply, or from capacitive coupling of the signal output to the input.

Someone versed in analog audio circuit design will consider adding a ground plane in the form of, for example, a third layer in a printed circuit board to keep input and output coupling to a minimum,

Feedback Motion Control

The Old Way

- 1) Build robot
- 2) Guess PID coefficients
- 3) Test
 - 3a) Express disappointment
 - 3b) Search Internet, modify PID values
 - 3c) Read book, modify PID coefficients again
 - 3d) Decide performance is good enough
 - 3e) Realize it isn't
 - 3f) See if anyone just sells a giant servo
 - 3g) Express disappointment
 - 3h) Re-guess PID coefficients
 - 3i) Switch processor
 - 3j) Dust off old Differential Equations book
 - 3k) Remember why the book was so dusty
 - 3l) Calculate new, wildly different PID coefficients
 - 3m) Invent new, wildly different swear words
 - 3n) Research fuzzy logic
 - 3o) Now it is certainly not working in uncertain ways
 - 3p) Pull hair
 - 3q) Switch controller
 - 3r) Re-guess PID coefficients
 - 3s) Switch programming language
 - 3t) Start a new project that doesn't need feedback control
 - 3u) See parts in box. Feel guilty. Go back to old project
 - 3v) Start testing every possible combination of PID coefficients
 - 3w) Apply eye drops to end, library, sleep-deprived eyes
 - 3x) Wait, it's working!
 - 3y) Decide not to do any more projects that require control systems
 - 3z) Wonder why someone doesn't just make a thing that tunes itself

The Kangaroo x2 Way

- 1) Build robot
- 2) Press Autotune
- 3) Get a snack

Kangaroo x2 adds self-tuning feedback to SyRen and Sabertooth motor drivers.



\$24.99

DimensionEngineering

www.dimensionengineering.com/kangaroo

NUTS & VOLTS

EVERYTHING FOR ELECTRONICS

Published Monthly By
T & L Publications, Inc.

430 Princeland Ct.
Corona, CA 92879-1300

(951) 371-8497

FAX **(951) 371-3052**

Webstore orders only **1-800-783-4624**

www.nutsvolts.com

Subscriptions

Toll Free **1-877-525-2539**

Outside US **1-818-487-4545**

P.O. Box 15277

North Hollywood, CA 91615

FOUNDER/ASSOCIATE PUBLISHER

Jack Lemieux

PUBLISHER

Larry Lemieux

publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING

Robin Lemieux

display@nutsvolts.com

EDITOR

Bryan Bergeron

techedit-nutsvolts@yahoo.com

CONTRIBUTING EDITORS

Jeff Eckert	Russ Kincaid
Joe Pardue	Fred Eady
Ron Hackett	Lou Frenzel
Ronald Anderson	Ron Newton
Matt Bates	John Gavlik
Martin Hebel	

CIRCULATION DEPARTMENT

subscribe@nutsvolts.com

SHOW COORDINATOR

Audrey Lemieux

MARKETING COORDINATOR WEBSTORE

Brian Kirkpatrick

sales@nutsvolts.com

WEB CONTENT

Michael Kaudze

website@nutsvolts.com

ADMINISTRATIVE ASSISTANT

Debbie Stauffacher



PRODUCTION

Sean Lemieux

Copyright © 2013 by T & L Publications, Inc.

All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: 430 Princeland Court, Corona, CA 92879.

Printed in the USA on SFI & FSC stock.  

and ample use of bypass capacitors on the microcontroller power input leads and analog components. They might also consider using a pre-amp and dedicated high resolution/low noise A/D converter chips instead of the relatively modest A/D circuitry in the microcontroller.

So, from a practical perspective, how do you get out of your technological comfort zone? One way is to scan through this issue of *Nuts & Volts* and try your hand at a project that you'd typically skip because it's not in your area of expertise.

The best way, of course, is to team up with someone who is both comfortable in his or her technological area and uncomfortable in yours. You'll both expand your comfort zones and, as a bonus, get a bit of practice mentoring — a great skill to develop. **NV**

READER FEEDBACK

Shaky Ground

I just wanted to pass this along to Ron Newton. We have been running his Gravimeter project from the January 2013 issue for the past few months and it appears to be working pretty well. I just pump the analog signal into my Arduino device and measure the signal via LabView.

I attached a DC jack to the unit so I can run it for weeks. Also, I notice on power-up, if I let the unit stabilize for a few hours I don't see as many spikes on the waveform. Do you think the spikes are due to hardware stabilization or earth movement at certain times of day? It's kind of interesting.

Rob Short W7FJ

Thanks for the feedback. I see these types of spikes on my seismographs all the time in the summer when it gets hot. Since they sit on a concrete floor, I have always assumed that it was the expansion and contraction of the concrete and its aggregate, or just the ground. However, with the gravimeter it may be the substrate settling down. If you get interested in seismology, take a look at the May 2012 issue. Recently on eBay, a batch of geophones were on sale for \$12 which really makes the unit very sensitive, small, and compact.

Ron Newton

Water Works

I have just read Don Hicke's feedback, "Fluid Remarks" in the August 2013 issue, and tend to disagree with his theory of the flux remover penetrating the stripboard and making it conductive. Those boards are dense and won't let normal cleaning fluids penetrate them. Additionally, I have never seen or used a flux remover that is conductive and, in fact, most of them could be sprayed on an operating circuit with minimal effect.

What I suspect is that Don was bitten by one of my old nemeses — water soluble flux from the solder. Water soluble flux leaves a coating on the board that can only be removed with water — not chemical cleansers. This coating that can't be seen by the naked eye is essentially salts that attract moisture and become conductive, as well as corrosive.

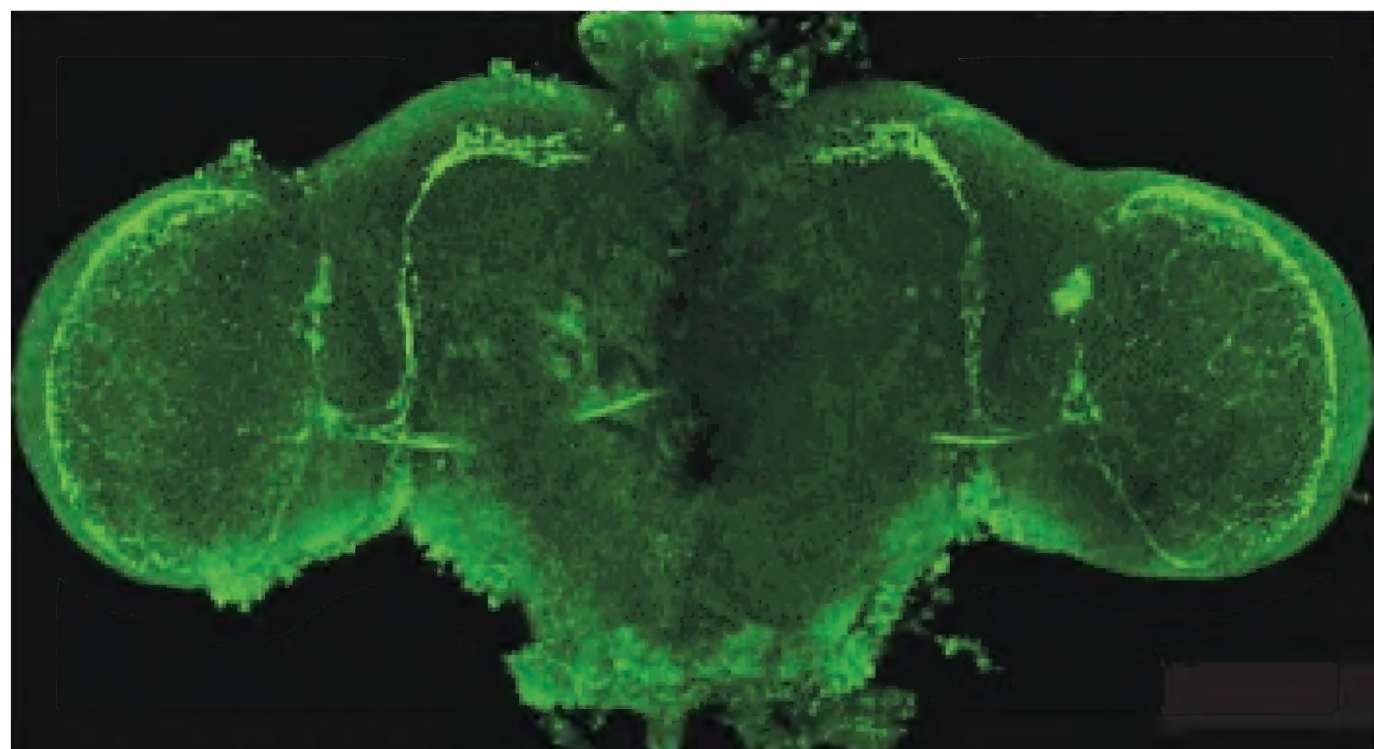
High humidity is enough to cause this effect, and the higher the circuit resistance or impedance, the more sensitive the circuit is to the leakage between components. If a circuit is assembled with a water soluble solder, it needs to be washed in clean hot water; preferably de-ionized if available.

Water washable flux is fine for a

Continued on page 52

Go to www.nutsvolts.com/index.php?/magazine/article/october2013_TechKnow13 to post comments about these topics.

ADVANCED TECHNOLOGY



■ The ArcLight protein provides a look at brain cell activity in a fruit fly.

Brain Sparks, Live

It is well known that the nervous system uses electrical activity for sensing what's going on around us, controlling bodily functions and just plain thinking. Naturally, doctors and other researchers have long been eager to find ways to monitor and control what's going on in our heads. Doing so generally involves the use of electrode banks, the use of voltage-sensitive (and highly toxic) dyes, or by detecting a surge of calcium ions that appear when a neuron receives a signal. In fact, progress has been made in developing genetically encoded calcium indicators (GECIs) which allow the calcium ions to fluoresce, offering a visual image of brain activity. This is an indirect process, however, as detecting calcium is not the same thing as detecting the voltage sparks themselves.

Now, some scientists at Yale's School of Medicine (medicine.yale.edu) have come up with a variation called genetically encoded fluorescent voltage indicators (GEVIs) that allow researchers to watch the sparks directly in a live animal.

As reported last August in the journal *Cell*, their "ArcLight" also allows researchers to see electrical activity in portions of the brain that were previously inaccessible. So far, the indicators have been used only in fruit flies, but the paper suggests that ArcLight and other GEVIs may prove useful in many ways for mapping brain cell activity in both normal and diseased animals, including us humans. ▲

A Boost for Low Carbon Vehicles

Despite considerable promotion and a slew of available tax incentives, the Electric Drive Transportation Association (www.electricdrive.org) reports that hybrid, plug-in hybrid, and all-electric cars still account for less than four percent of the US market, with total cumulative sales only recently passing 115,000. Their drawbacks are no secret: high purchase price, limited range, and (in the case of all-electrics) long charging times. Fortunately, a recent development at Britain's National Physical Laboratory (www.npl.co.uk) could be a significant step toward solving some of those problems.

The power conversion and management systems in automotive power electronics depend not only on the characteristics of the batteries, but also on capacitors which provide energy bursts for climbing hills, and can absorb energy otherwise wasted on braking. Because typical capacitors lose capacitance at high temperatures, automotive power electronics require quite a bit of cooling which adds to vehicle weight. NPL notes, for example, that barium titanate caps can lose as much as 85 percent of their capacitance at working voltage.

The lab has developed a new ceramic capacitor dielectric material — dubbed HITECA — that offers a higher energy density and operates with stable capacitance at temperatures as high as 200°C (392°F). With its high permittivity and reduced capacitance loss, caps based on it could enable smaller electronic devices and improve vehicle performance.

In addition, "HITECA capacitors could improve high temperature electronics in the aerospace, power, oil, and gas sectors, and in high energy applications such as pulsed power where energy is stored over a period of time before being released as a high power pulse." ▲



■ New high temperature, high capacitance device may improve EV performance.

COMPUTERS and NETWORKING

Gaming Laptop

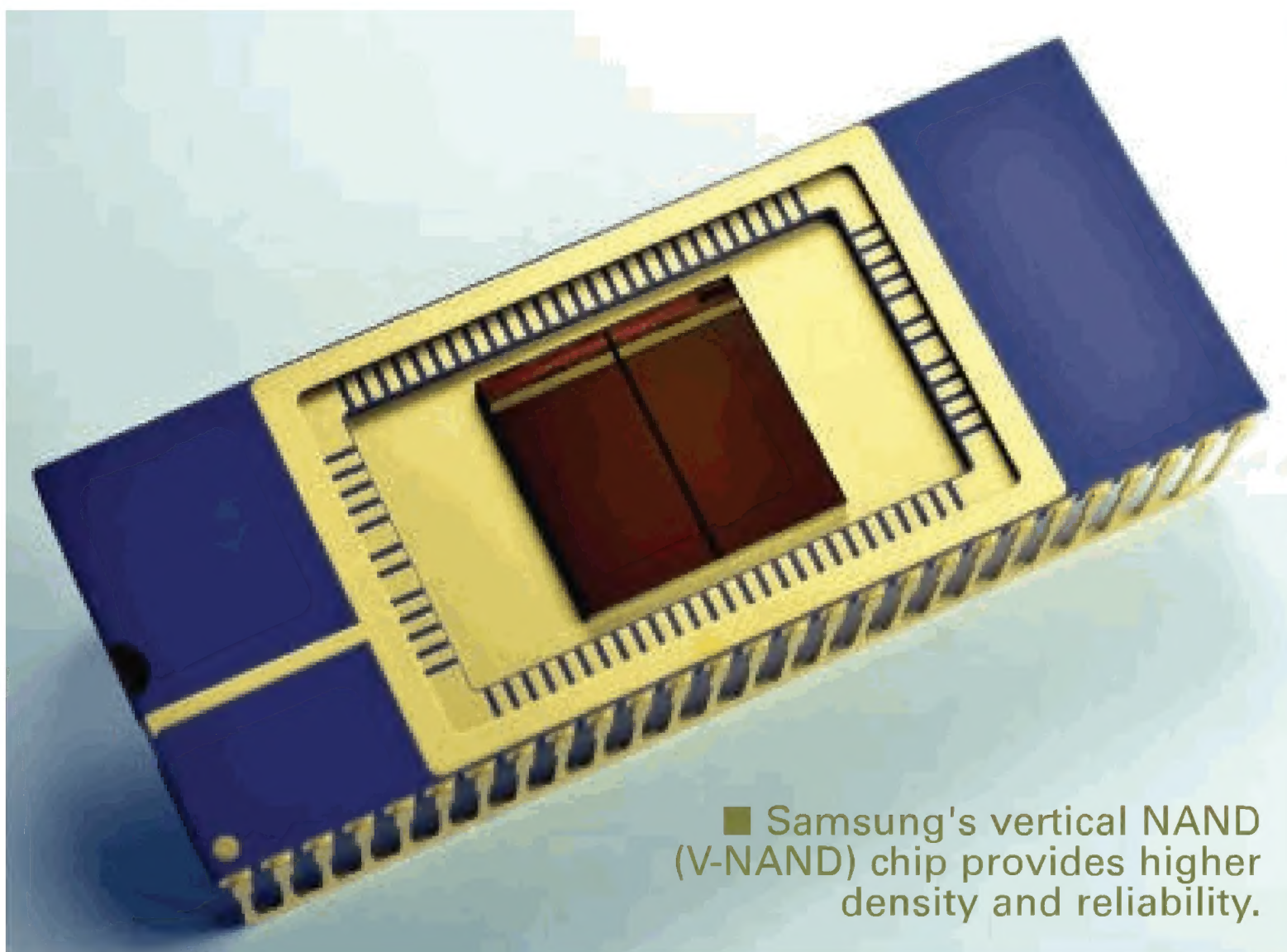
If you're looking for a compact laptop but also want to immerse yourself in gaming, you might take a look at the GE40 — a recent introduction from MSI Computer Corp. (www.msimobile.com). Billed as the "perfect combination of ultrabook portability and deadly gaming capability," it provides graphics via a NVIDIA GeForce GTX 760M card enhanced by a matrix display that allows output to three displays at once (two external). It also includes Sound Blaster Cinema and Audio Boost technologies to deliver high quality surround sound. The GE40 weighs only 4.4 lb (2 kg) and is less than one inch (2.5 cm) thick, so you can play "Blade and Soul" anywhere you go.

Other specs include a Core i7-4702MQ processor (2.9-3.2 GHz), a 14 inch HD display, 8 GB of 1,600 MHz DDR3, and Giga LAN 10/100/1000. It will set you back



■ MSI's GE40, geared for portable gaming.

either \$1,299 or \$1,399, depending on your choice of drives. ▲



■ Samsung's vertical NAND (V-NAND) chip provides higher density and reliability.

3D Memory in Production

One of the latest developments in memory technology is the 3D NAND Flash which extends the standard 2D planar technology by turning the chip to a vertical position and then stacking a bunch of them side by side to create a higher density chip. This results in about double the write performance and much higher reliability.

In August, Samsung Electronics (www.samsung.com) announced the industry's first mass production of the devices. The company's V-NAND offers 128 Gb density on a single chip. Using Samsung's 3D Charge Trap Flash (CTF) technology and a special interconnect technology to link the 3D array, it provides twice the scaling of a standard 20 nm class planar device. Since up to 24 layers can be stacked vertically, Samsung will be able to create even higher capacity chips. This has "set the foundation for "more advanced products,

including one terabit (Tb) NAND Flash ..." The Samsung press release did not name any companies or specific devices that will be employing the chip, but "mass production" presumably means they will be turning up somewhere soon. ▲

USB Speeds Up

It seems like only yesterday that the USB 3.0 standard turned 2.0 devices into relative slugs, but in reality the standard was released back in 2008. In July of this year, the USB Implementers Forum upped the ante with SuperSpeed USB 3.1 — a spec that adds enhancements to enable operation at up to 10 Gbps. Fortunately, your existing devices are safe from complete obsolescence

given that "compatibility is assured with existing USB 3.0 software stacks and device class protocols, as well as with existing 5 Gbps hubs and devices, and USB 2.0 products." The USB Promoter Group includes Hewlett-Packard, Intel, Microsoft, Renesas Electronics, ST-Ericsson, and Texas Instruments, so we should be seeing compatible devices appearing soon. For details or to download a PDF of the spec, visit www.usb.org. ▲

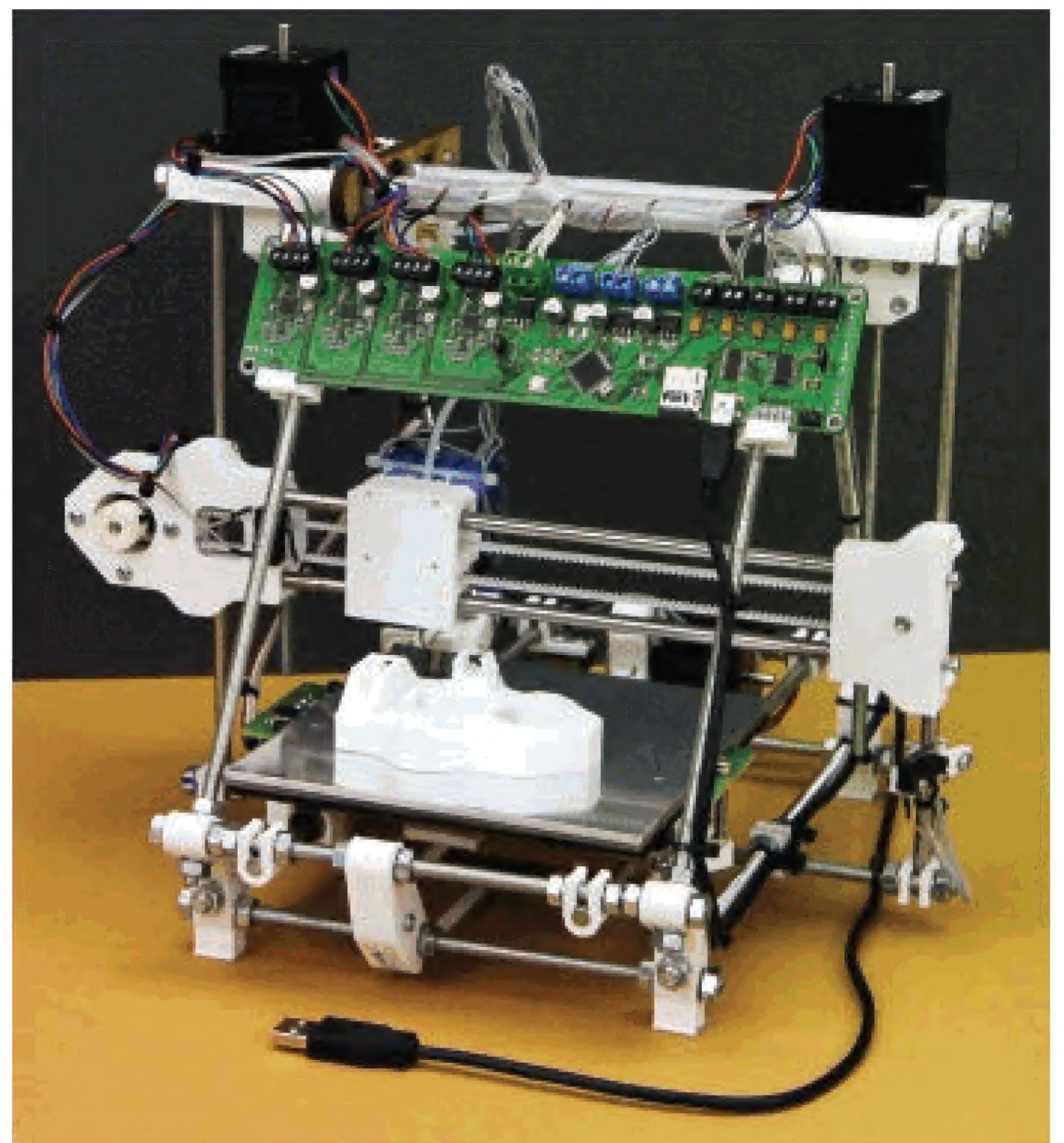
CIRCUITS and DEVICES

Free 3D Printer — Maybe

3D printers remain a hot item, and they are becoming more practical every day. In fact, a Michigan Technological University researcher is slated to publish an article in the journal *Mechatronics* documenting how he chose 20 common household items (cell phone accessories, a garlic press, a showerhead, a spoon holder, etc.) that would cost between \$312 and \$2,000 to buy retail, and created them for only \$18 with a 3D printer. (For a pdf of the paper, just aim your browser at www.jkeckert.com/freedownloads/3D.pdf.)

The problem is that the machines are still not cheap. A MakerBot Replicator 2, for example, will run \$2,799, and you can buy a lot of premade plastic paraphernalia for that kind of money. In theory at least, it now is possible to get one for free — or close to it. The catch is that you have to know someone who already has a RepRap (replicating rapid prototyper) — a unit specifically designed to be "humanity's first general-purpose self-replicating manufacturing machine." Otherwise, you'll have to build one yourself.

You begin by logging onto www.reprap.org — home of the RepRap Project. Founded in 2005 by a mechanical engineering lecturer at the University of Bath, the organization so far has released six open source designs ranging from the full-sized Prusa Mendel model down to the smaller, simpler RepRapPro Huxley. The site provides



■ The RepRapPro Huxley, a portable self-replicating 3D printer.

info about construction, where to obtain parts, and where you can buy kits and even assembled units. More than 50 vendors worldwide are listed, and the list is not intended to be complete. When yours is up and running, you can even become a vendor yourself. ▲

Wireless Camera Memory

These days, many people rely on a smartphone's built-in camera for most of their photographic needs. The quality is decent, and it's nice that you can immediately share vacation pics with anyone else, anywhere in the world. Some of us still prefer the added quality and features of a dedicated digital camera, such as optical zoom, faster speeds, low light operation, better resolution, interchangeable lenses, and so on. The drawback is that even

your \$6,000 Nikon D4 can't share images wirelessly unless you buy the optional \$877 WT-5A transmitter. You can, however, add that capability easily to many cameras (alas, not the D4).

All you have to do is plug in an Eye-Fi Mobi wireless memory card. The device includes built-in Wi-Fi, so you don't even need a network, hotspot, or Internet connection to use it. You just transfer files directly to another device. The retail price for an 8 GB card is \$49.99, and \$79.99 for the 16 GB one. ▲



■ Send photos directly from your digital camera to smartphones and tablets.

Digitizing the Throne

With everything else in the average household becoming wired, digitized, and automated, it was only a matter of time until plumbing fixture manufacturers came up with a super high-tech toilet. It looks like a Japanese company — INAX (www.inax-usa.com) — has become Lord of the Loo with its Regio line, incorporating "every technology possible today, including the highest levels of water conservation and automated convenience."

The Silent Stream flush unit is powered by an air-drive mechanism, so all you'll hear is the sound of a "murmuring brook." The lid opens and closes automatically, and plays digital music via remote control while you inhabit the throne. INAX's Plasmacluster® technology takes care of disagreeable odors, and the bowl is even illuminated with LEDs so you can make sure everything comes out all right. Of course, the unit is equipped with both front and rear cleansing nozzles offering a choice of spray modes, including a massage feature.

The white model comes in at \$6,903, but you'll no doubt want to upgrade to black for only about \$1,700 more. (There is a gold-plated model for about \$40,000 but, after all, you wouldn't want visitors to think you're pretentious.) Unfortunately, none of these versions include the remote control stand which will run another \$595. It all adds up to such an elegant experience that you'll be making extra stops at Taco Bell just to encourage nature to call. ▲



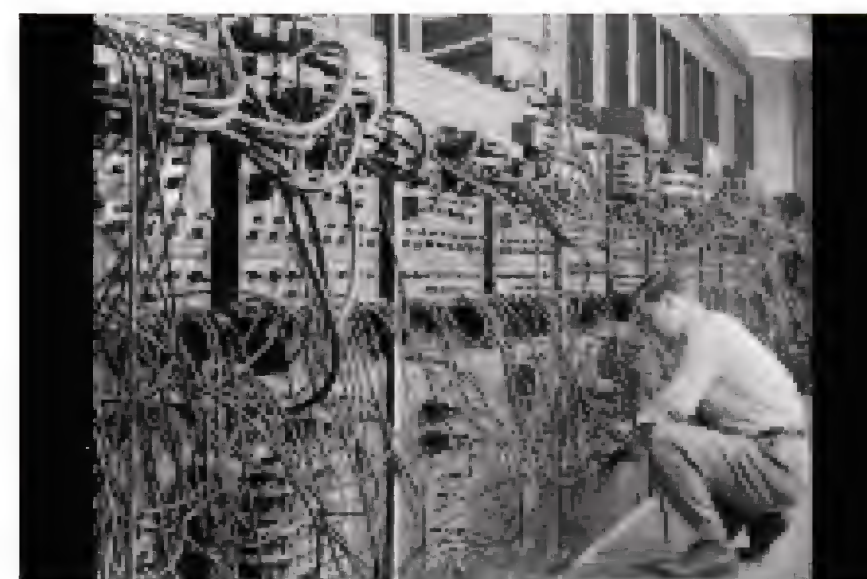
■ INAX's Regio high-tech toilet: New comfort for the modern age."

INDUSTRY and the PROFESSION

ENIAC Turns 70

It was back in 1943 when work began on the Electrical Numerical Integrator and Computer (ENIAC) at the University of Pennsylvania's Moore School of Electrical Engineering, under the direction of chief engineer J. Presper Eckert and chief consultant John Mauchly. Thus, my dear old Uncle Presper (just kidding) initiated construction of the world's first electronic general-purpose computer, which was not revealed to the public until 1946. Designed to calculate artillery firing tables for the US Army, it was 1,000 times as fast as electromechanical calculation devices of its day. The ENIAC cost about \$6 million in today's money, which isn't bad for something that's 3 x 8 x 100 ft, weighs 27 tons, and draws 150 kW. That's quite a bit of power, but it takes juice to run 17,468 vacuum tubes, even though several of them burned out on a daily basis, leaving the machine nonfunctional about half the time.

The machine was capable of performing up to 5,000 instructions per second, assuming those instructions were additions. The number dropped to 357 for multiplications and 38 for divisions. (Today's Intel Core i7 processors — for comparison — can do 82.3 billion per second when running at 2.66 GHz.) The machine was shut down forever at 11:45 p.m. on October 2, 1955. Pieces of it are still on display in various museums. **NV**



■ The ENIAC computer circa 1946.

Getting Started with Python Programming

In this month's installment, we're going to begin our Raspberry Pi (RPI) programming experiments by implementing a simple "Hello World!" program using Python, and then move on to the basics of controlling the RPI's GPIO pins with a Python program. However, before we get to that, I want to briefly discuss my reasons for choosing Python as the programming language for our experiments. So, let's get started!

Why Python?

A few months ago when I first ordered my RPI, I began searching the Web for information about the various programming languages that can be installed on it. Each time I discovered another RPI programming language, I added it to a list I was maintaining. Before long, my little list had more than 50 entries.

Just to give you an idea of the scope of the programming possibilities for the RPI, here's a list of a dozen compatible languages that I think might be the most easily recognized: BASIC, C, C++, Forth, Java, JavaScript, Pascal, Perl, PHP, Python, Ruby, and Scratch.

With so many languages from which to choose, you may wonder why we're starting with Python. So, I'll explain this decision. To begin with, Python is free; it's already installed on the Pi; and it's the "official" language for the Pi. (In fact, the "Pi" in "Raspberry Pi" is actually a reference to Python.) As a result, it's probably the most popular programming language for the RPI, which means that the online support is outstanding.

Also, Python is relatively easy to learn, read, and understand. Much easier, for example, than C or C++ — both of which are also very popular on the RPI. Python includes a large standard library containing many built-in code modules that greatly simplify programming various complex tasks. (In order to use the functions in a built-in module, we first need to import that module into our program — which we'll do before we're finished this month.)

It's true that both C and C++ contain even more built-in modules, but these modules can be accessed by Python, as well. In addition to Python's built-in modules, there is also a wide variety of user-developed external modules that can be easily imported into Python when they are needed. (Later in this article, we will do this, as well.)

When I first began experimenting with the RPI, I had absolutely no experience with Python, so I naturally had dozens of questions along the way. I quickly discovered that I could always find an answer online to my question. Being somewhat old-fashioned, I ordered a couple of RPI-related books that turned out to be very helpful. You can refer to the **Resources** sidebar for some

suggestions. (If you're just starting out with Python programming, I think the best place to start is the **Python.org** Python 3 Tutorial.)

Another major advantage of using Python is that there are free versions that can be installed on Macintosh OS X, Linux, and Microsoft Windows computers, as well as the iPhone, iPad, and Android phones and tablets. This cross-platform flexibility means that we can use Python to develop PICAXE-Pi projects that include the display and/or manipulation of data on just about any modern computer, tablet, or cell phone. One of the goals I have in mind for our PICAXE-Pi projects is to be able to use a cell phone to monitor and update a running PICAXE home automation project.

The final reason for choosing Python may be a little controversial. I love this feature of Python, but some readers may not like it at all: Python may be the only language that "forces" programmers to appropriately indent their source code!

If you fail to do so, you will either get a syntax error or a series of (somewhat cryptic) errors. Later in this article, we'll take a look at how indentation works in Python. For now, however, let me explain why I love this feature. As an illustration, let's consider two versions of a very simple PICAXE BASIC code snippet:

Version 1

```
for index = 1 to 20
  if index // 4 = 0 then
    high LED
  else
    low LED
  endif
  wait 1
next index
```


Version 2

```

for index = 1 to 20
  if index // 4 = 0 then
    high LED
  else
    low LED
  endif
  wait 1
next index

```

Both code snippets implement the same task — which could also be accomplished with simpler code — but that’s not the point. Rather, the question is which snippet is easier to read and understand? (If you chose Version 2, you’re absolutely correct!)

If you’re a regular Primer reader, you know that I often encourage readers to email me with any PICAXE-related questions or problems they may be having. I frequently receive software questions accompanied by a program listing. Sometimes (fortunately not often) the listings look a lot like Version 1 above. Before I can even begin to try to figure out what’s going on, I have to edit the program so that it’s properly formatted and more easily readable.

With Python programs, I won’t need to do that because Python forces us to correctly format all our programs. That’s why I love this feature! (There’s another major benefit to formatting a Python program; we’ll discuss that one when we get to our “Hello World!” stuff.)

Python 2 vs. Python 3

Python has been around for almost 20 years, and it has evolved into a very popular and easy to use language. However, when Python 3.0 was introduced in 2008, it created quite a stir in the Python community because it was the first version that wasn’t “backwards-

compatible.” In other words, some of the changes in Python 3 caused programs written in earlier versions of Python to produce syntax errors.

As a simple example, if we wanted to display “Hello World!” on the RPi monitor (which we will shortly do), in Python 3 we would write ***print*** (“Hello World!”); in all earlier versions of Python, we would write ***print*** “Hello World!”

Understandably, many Python programmers were somewhat upset by this situation. However, others welcomed the changes because they make Python more consistent and more readable. (Search for “Python 2 vs. Python 3” if you’re interested in the details of the debate.)

Of course, we have to make a choice at this point: Are we going to use Python 2 or Python 3 for our PICAXE-Pi experiments? I have to admit, I’m having a little trouble deciding!

When I first started experimenting with Python, I chose to use version 3.2 because I knew the Python community was already moving in that direction, and I didn’t want to end up trying to learn both versions, probably getting confused in the process. At that point, I intended to use I²C to communicate between the PICAXE and RPi, but I

quickly discovered that an I²C library was not yet available in Python 3.2, so I switched to using Python 2.7.

After experimenting with I²C for a few days, I realized that it’s fairly complicated to implement reliable two-way communications between the two processors. It turns out that good old-fashioned serial communication looks like a better choice. (We’ll get into those details in a future column.)

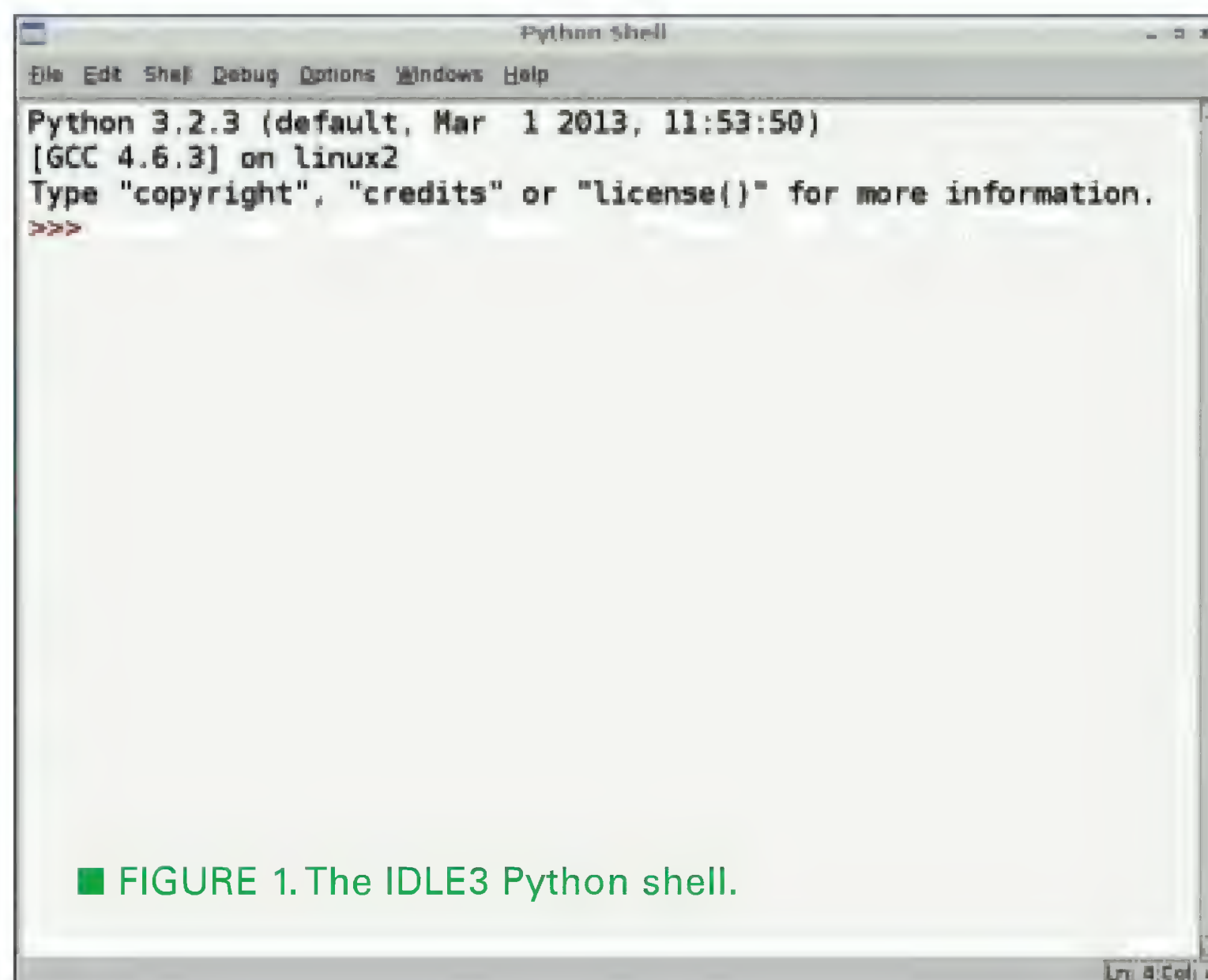
The point of all this is that I’ve switched back to using Python 3.2. Naturally, I can’t rule out the possibility that I won’t run into additional problems with Python 3. Fortunately, however, Python 2.7 and Python 3.2 are both pre-installed in the latest version of the RPi’s operating system, so switching between the two versions won’t be a problem.

Experimenting With Python 3.2

When we program in PICAXE BASIC, we use either the Programming Editor or AXEpad software — both of which are PICAXE IDEs (Integrated Development Environments). Similarly, there are several IDEs for Python programming.

In the Python world, an IDE is usually referred to as a Python shell, and the shell that’s included in the RPi operating system is called IDLE. Actually, when you look at the RPi desktop, you will see two different IDLE icons. IDLE is used for Python 2.x programming, and IDLE3 is used for Python 3 programming.

On your RPi, double-click the IDLE3 icon and you will see a window that’s similar to the one shown in **Figure 1** which is a screenshot from my



■ FIGURE 1. The IDLE3 Python shell.

RPi. As you can see, I'm running Python 3.2.3, but any recent version of Python 3 will be fine for our purposes.

Both versions of the IDLE shell can be used in two different ways: Single Python commands can be executed immediately by typing them at the triple chevron (`>>>`) interactive prompt, or a complete program (a.k.a., a script) can be written in a separate window. Then the entire script can be executed.

Let's begin our Python explorations with the simpler interactive approach, and test a few of Python's math skills. At the shell's interactive prompt, type `2+2` and press the enter (or return) key. Python will respond with the correct answer, and provide you with another prompt. Actually, Python is very good at math — try this one: `2+3*5`. If you remember the "order of operations" from high school Algebra, you won't be surprised by the answer.

Here's another one to try: `(2+3)*5`. Obviously, Python also knows how to handle parentheses in an expression. Let's push the envelope: `458.39*12.005`. If you have a calculator handy, you can check Python's accuracy.

If you want to stump Python, let's try to calculate the circumference of a circle that has a radius of 3, so type the following at the interactive prompt: `2*pi*3`. When you press the enter key this time, you will discover that Python doesn't know the value of pi. (Irony, isn't it?)

However, as a simple demonstration of the power of Python's built-in library, type the following at the interactive prompt: **from math import pi**. This statement

doesn't import the entire math module; it just imports the constant **pi**. When you execute the statement, nothing seems to happen, but now try `2*pi*3` again; this time, Python gets it right! In addition to the value of pi, the math library includes all sorts of trigonometric functions, logarithms, and a lot more stuff that you may or may not want to know about. If you are curious, take a look at section 8.2 of the standard library documentation (see the **Resources** sidebar) and you will see what I mean.

```
print("Hello World!")
print("Hello World!")
print("Hello World!")
print("& Hi Python!")
```

When you have finished typing the program, save it as "Hello" – the .py extension will be automatically added to the name in order to identify it as a Python program. Actually, saving the file at this point is optional. When we run the program (which we're about to do), the shell forces us to save it first anyway. Either way, just select the Run menu's Run

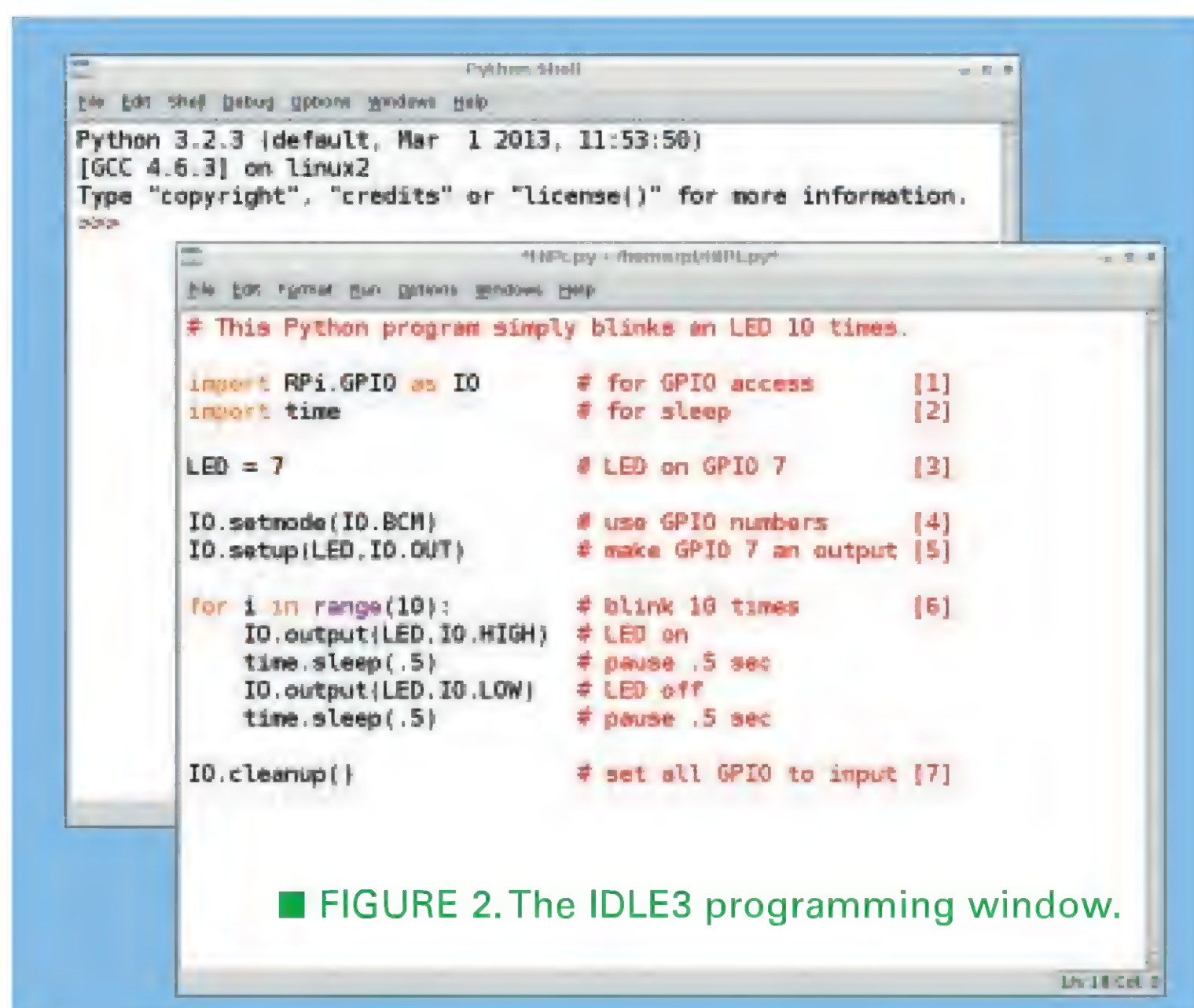
Module option (or, simply press the F5 key) to run the *Hello.py* program, and observe the program's output in the shell. Of course, *Hello.py* is a fairly trivial program, but it did give us a chance to get comfortable with using the shell to write a Python program.

Before we move on to the main goal of this month's Primer (using Python to control the RPi's GPIO pins), let's jazz up our little Hello World! program by

using a simple version of the Python *while* loop. In PICAXE BASIC, one way to execute the same code multiple times would be to use a *do/loop*. For example, in PICAXE BASIC, if we wanted to execute the same code 10 times, we could write the following:

```
index = 1
do while index < 11
    ' some code goes here
    index = index + 1
loop
' program continues here
```

Of course, we probably wouldn't really use a PICAXE *do/loop* for this



■ FIGURE 2. The IDLE3 programming window.

"Hello World!" Program

When you have finished experimenting with the interactive prompt, we can move on to writing our first real Python program. To accomplish this, we first need to open a new window in which we can write, edit, and save a program before running it. In the shell window, click on the File menu, and choose the first option (New Window) which opens a second window in which we can write our program (see **Figure 2**).

Type the following code into the new window:

purpose; a *for/next* loop would be simpler. However, in Python, the reverse is true; a Python *while* loop is simpler than a Python *for* loop, which is why we're using the *while* loop for our first example of looping in Python. In Python, a # symbol is used to indicate a comment, so the equivalent loop could be written as follows:

```
index = 1
while index < 11:
    # some code goes here
    index = index + 1
# program continues here
```

There are three visible differences between the two loops; let's examine them in the order in which they appear:

1) PICAXE includes do – Python does not. This is largely stylistic; the syntax of either language could have included or not included the *do*.

2) PICAXE does not include a colon – Python does. In Python, a colon is **required** before any indented code. As I mentioned earlier, Python also **requires** the indentation. Of course, I indented both snippets because structured code is easier to read, but PICAXE BASIC **does not** require it.

If I had lined up the left edge of every line of BASIC code, the first snippet would run correctly. On the other hand, left-justifying all the lines in the Python code will result in an error, and the shell will refuse to run the program.

3) PICAXE includes a loop statement – Python does not. The fact that unstructured code is allowed in PICAXE BASIC means that we need some other way to indicate where the loop ends, so the *loop* statement is required. The Python interpreter, on the other hand, knows that the loop ends when the indentation ends, so we don't need the additional statement.

To my way of thinking, it's the best of both worlds; Python demands

structured code (which is easier to read and understand), and it reduces the number of lines in a program. You may disagree, but you will obey!

There's a fourth significant difference that isn't at all visible in the two snippets. In PICAXE BASIC, we need two separate statements to declare a variable (e.g., **symbol index = b0**) and to assign a value to that variable (e.g., **index = 1**). If we forget to include the *symbol* statement earlier in the program, we'll get a syntax error when we attempt to assign a value to the variable.

In contrast, Python allows us to declare a variable name and assign a value to that name in the same step (e.g., **index = 1**).

This is a good place to mention another feature of Python, which I didn't include earlier in the discussion of Python advantages because I'm not sure that it is an advantage. However, it's definitely important to keep this feature in mind: Unlike PICAXE BASIC, **Python is case-sensitive**. If you accidentally change the case of even a single letter in a Python reserved word, your program will produce a syntax error and fail to run.

Python's case sensitivity is even more significant when it comes to variable names. If you write **myVar = 2** at one point in a program, and later try to update the value of that variable by accidentally writing **myvar = 10**, you have actually created a second variable — not updated the **myVar** variable as you intended. Even worse, you won't get a syntax error; your program will just behave unpredictably!

With that background information in mind, let's edit our *Hello.py* program so that it includes a *while* loop:

```
index = 1
while index < 11:
    print("Hello World!")
    index = index + 1
print("& Hi Python!")
```

When you've made the necessary changes, save the modified program as *HelloLoop*, and run it. (Don't forget, the Python shell will automatically add the .py suffix.)

Before moving on to the topic of controlling the RPi's GPIO pins with a Python program, take some time to experiment with changing the indentation of the *HelloLoop.py* program to see the variety of syntax errors you can produce.

For example, make each of the following changes and run the program again for each change:

- Delete the colon.
- Put the colon back and "outdent" the **print("Hello World!")** statement, so that it's left-justified with the **while** statement.
- Return the **print("Hello World!")** statement to its correct position and indent **print("& Hi Python!!")**.

Warning: It is possible to make some changes that can result in an infinite loop. If that happens, try pressing *ctrl-c* to exit the program. If that doesn't work, try quitting the shell. As a last resort, rebooting your RPi should solve the problem.

When you're finished wreaking havoc on the Python interpreter, we can move on to our final topic!

Hi-Pi! Program

Now that we're familiar with writing a simple Python program, we're ready to tackle the task of controlling the RPi's GPIO pins.

For this purpose, I'm using the stripboard circuit we constructed last time. However, the circuitry for our Hi-Pi! program is very simple, so you can certainly use your own setup, as well.

All we need is an LED and a current-limiting resistor (which is already included in our stripboard circuit) connected between GPIO pin 7 and ground.

The hardware setup that I used is

shown in **Figure 3**. (You can't see it in the photo, but my RPi is still connected to the other end of the ribbon cable.)

Note that I'm using the two small stripboard connectors you saw last time to connect the stripboard to a breadboard. Of course, you can just use jumper wires for that purpose, if you prefer.

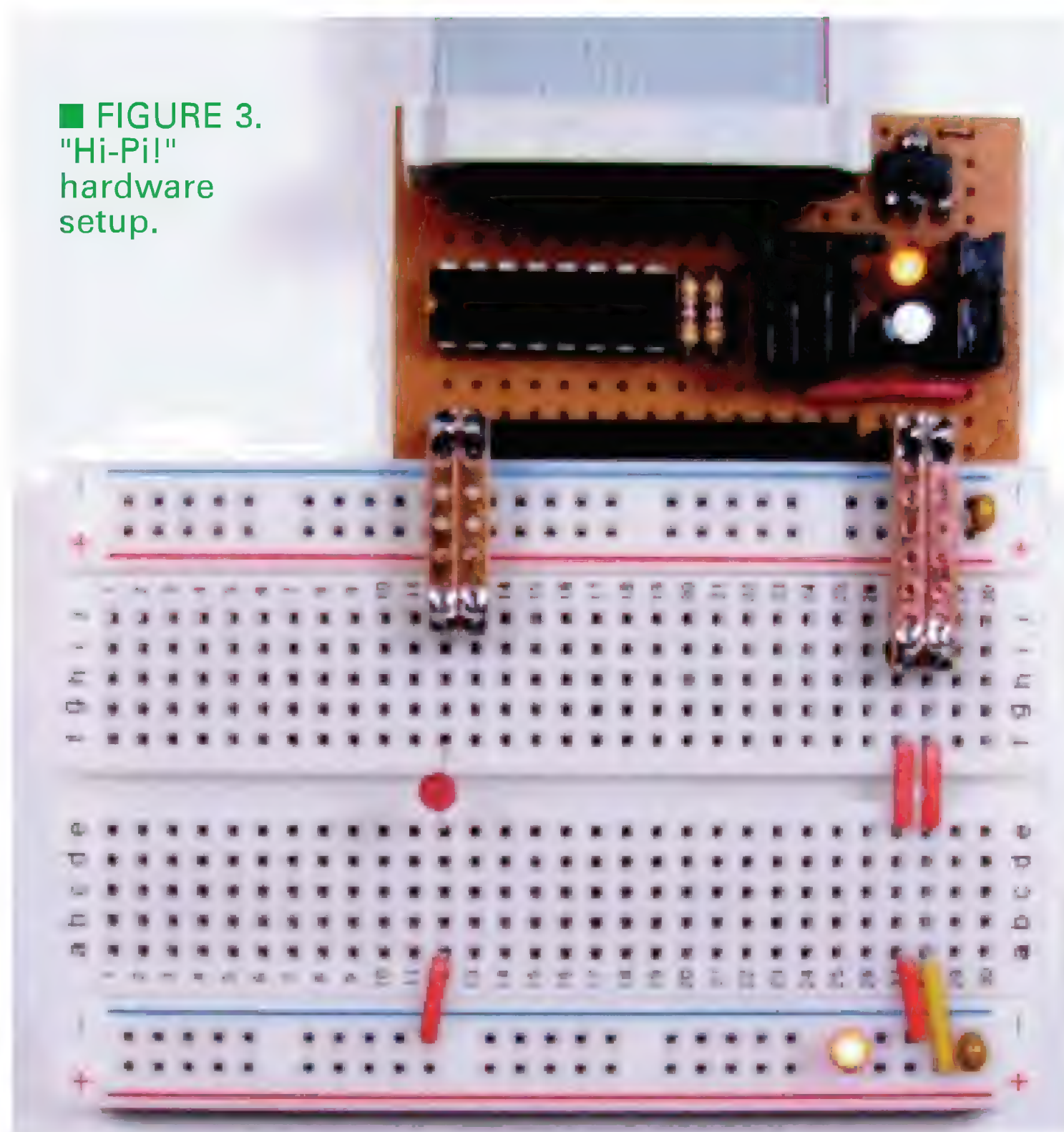
When you've completed your hardware setup, we can move on to discussing the Python program (*HiPi.py*) that we'll use for this experiment. (The program is available at the article link.) Of course, the easiest way to get the program into your RPi would be to use the Midori browser to download the file directly to

your RPi. However, at this point, I haven't been able to successfully unzip the *N&V* file on my RPi. (For some reason, it doesn't recognize the zip format.) I'll try to resolve this problem before the October column is available, but just in case I can't, I have also put the unzipped program file on my site (www.JRHackett.net/python.shtml) so that you can download it to your RPi. (You can't directly navigate to my Python page; just type the address into your browser's URL bar.)

Of course, there are other ways to move files from a Mac or PC to the RPi, but it would take too much space to discuss them here. The first method you may want to research is how to use a USB Flash drive with the RPi.

Not long after I began experimenting with my Pi, I managed to trash the operating system, so I had to

■ **FIGURE 3.**
"Hi-Pi!"
hardware
setup.



reformat my SD card. Of course, I lost all the Python programs I had written up to that point. It was a great motivator for learning how to back up my programs on a USB Flash drive.

The complete *HiPi.py* program is in **Listing 1**. As usual, in order to facilitate our discussion of the program, we'll use the numbers following some of the comments.

1) *RPi.GPIO* is one of several

an *RPi.GPIO* function would have to include the full name of the module. For example, in the line that includes the *LED on* comment, we would need to write **RPi.GPIO.output(LED,RPi.GPIO.HIGH)** which involves a lot more typing. Since the same is true for all the other programming lines that access an *RPi.GPIO* function, using an alias with the *import* statement can save us a considerable amount of typing.

2) In the second *import*

statement, the name **time** (which is a built-in module) is short enough, so I didn't bother with an alias.

3) This line looks just like a PICAXE BASIC statement, but don't forget that here we're both declaring the *LED* variable, and assigning the value 7 to it (because the LED is connected to GPIO pin 7).

```
# This Python program simply blinks an LED 10 times.

import RPi.GPIO as IO      # for GPIO access      [1]
import time                # for sleep             [2]

LED = 7                    # LED on GPIO 7          [3]

IO.setmode(IO.BCM)         # use GPIO numbers    [4]
IO.setwarnings(False)      # disable warning    [5]
IO.setup(LED,IO.OUT)       # make GPIO 7 an output [6]

for i in range(10):        # blink LED 10 times   [7]
    IO.output(LED,IO.HIGH)  # LED on
    time.sleep(.5)         # pause .5 sec
    IO.output(LED,IO.LOW)   # LED off
    time.sleep(.5)         # pause .5 sec

IO.cleanup()               # set all GPIO to input [8]
```

Listing 1.

4) In the *RPi.GPIO* module, the *setmode* function is used to determine the numbering mode for all I/O commands. *RPi.GPIO.BCM* (or, in this case *IO.BCM*) specifies the GPIO pin numbering; if you prefer to use the RPi I/O header's physical pin numbers (1-26), this line should read ***IO.setmode(IO.BOARD)***.

5) If you omit this line, you may get a "GPIO channel in use" warning as the program starts to run. The warning does not cause a problem; the program runs correctly. However, many people find the warning annoying, so they include this statement.

6) Here, we configure GPIO pin 7 as an output. In the next Primer, we'll see how to configure a pin as an input (but I bet you can already guess!).

7) This line implements a loop that executes 10 times. It involves two Python constructs that are new to us: the *range()* function and the *for* statement. However, in order to understand what's happening here, we need to back up and briefly discuss Python's *list* data type.

In Python, a list is one of several compound data types; it's a powerful data structure that can contain many different data values. A list can be written as "a list of comma-separated values (items) between square brackets." The items don't have to have the same data type, but this time we're just using integers. (See the **Python.org** Tutorial, section 3.1.4 for more detail.)

Here's a simple example of a Python list: *myList* = [2,4,6,8,10]. Items in a list are automatically indexed, starting at 0; e.g., *myList*[2] returns 6, because 2 is the index of the third item in the list.

The reason we need to discuss lists at this point is that the *range()* function returns a list (which is also zero-based). In its simplest

form, *range()* takes a single integer argument, and returns a zero-based list of integers. For example, *range*(10) returns the following list: [0,1,2,3,4,5,6,7,8,9]. (Details for the other — more powerful — forms of the *range()* function are available in section 4.3 of the **Python.org** tutorial.)

For our present purpose, the most important thing about lists is that they are iterable, which means that the values in a list can be used to control the number of times that a loop repeats, or iterates. As a result, the *for* loop repeats 10 times, with the variable *i* incrementing from 0 in the first iteration of the loop, through 9 in the last iteration of the loop. That's a total of 10 iterations, so the LED blinks 10 times.

Again, note that we didn't need to declare the variable *i*; that's automatic in Python. We also didn't use *i* for anything other than the loop control variable, but that's often the case in PICAXE BASIC, as well:

```
for index = 0 to 9
  gosub blink
next index
```

8) *Cleanup()* is another function that's included in the *RPi.GPIO* module. It resets all GPIO pins to inputs, with no pullup or pulldown resistors. (We'll discuss GPIO inputs and pullup/pulldown resistors next time.) Note that *cleanup()* does not take an argument, but from what I can gather, it's considered good programming practice to include the

empty parentheses anyway.

It's important to include the *cleanup()* function at the end of every program that uses the GPIO pins. Don't forget! Even when you turn off the power to your stripboard circuit, the RPi's GPIO pins remain active.

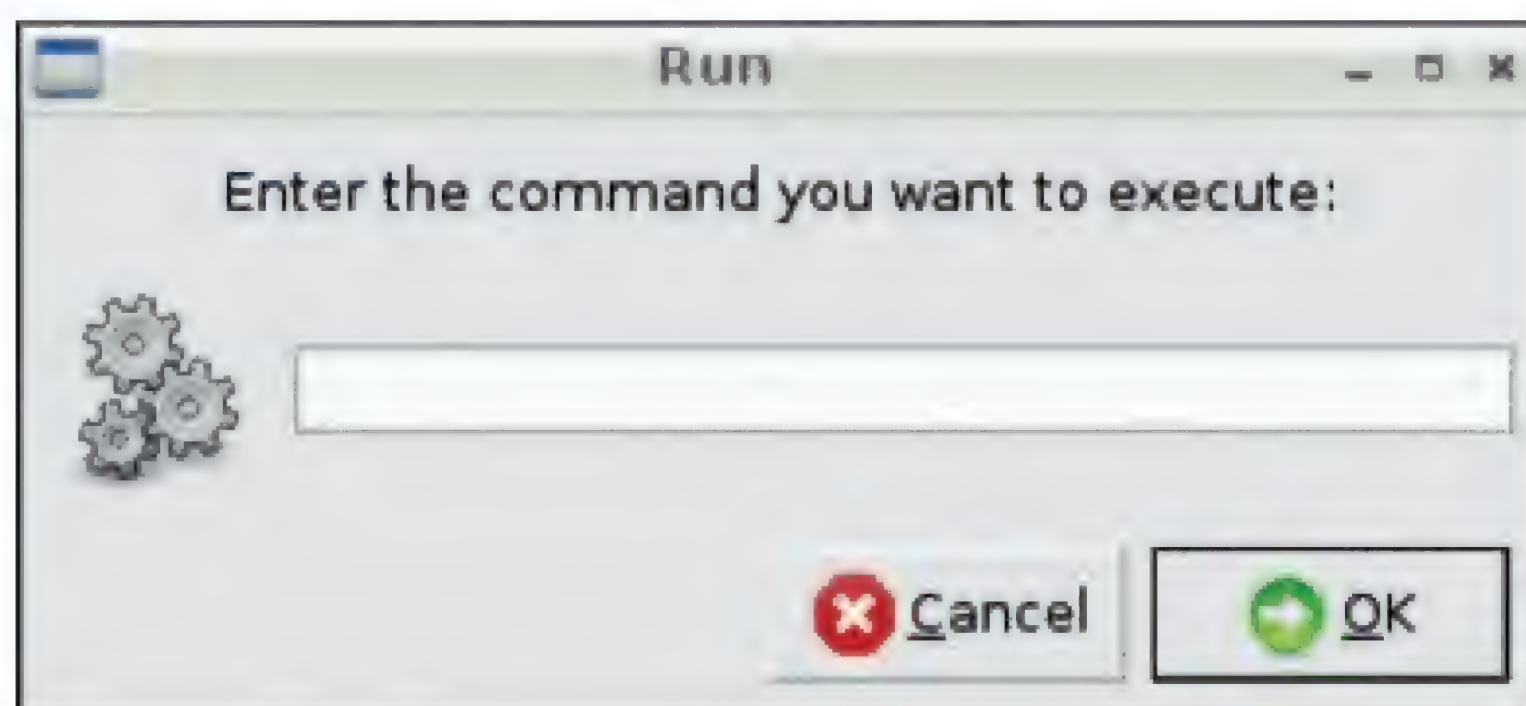
If you're modifying your hardware setup, any pin that's configured as an output when your program terminates presents the risk of an accidental short. Executing the *cleanup()* function minimizes the risk, but the only way to eliminate it entirely is to completely shut down your RPi and turn off the power.

Assuming that you've found a way to get the *HiPi.py* program onto your RPi (as a last resort, you could always type it into an IDLE3 programming window), we're almost ready to run it. In case you missed the "almost," we still have one more point to discuss before we actually run the *HiPi.py* program.

If you've been experimenting with your RPi, I'm sure you have come across the Linux *sudo* command which is a contraction of "super-user do." On a Linux system, the super-user (sometimes also referred to as the root user) is similar to the administrator on a Macintosh or Windows PC; he or she has special privileges that ordinary users don't have. On the Pi, accessing the GPIO pins is one of those special privileges. Therefore, if you run the IDLE3 shell and then open the *HiPi.py* program and try to run it, you will encounter the following error

message: *Runtime Error: No access to /dev/mem. Try running as root!*

The usual solution to this problem is to use the Linux terminal so that you can enter the necessary *sudo* command. To do so, quit IDLE3 if it's still running, open the terminal, and type the



■ FIGURE 4. The terminal's single command window.

following command: `sudo idle3`. (I have no idea why this command is lowercase, but the IDLE3 icon is uppercase. If anyone does, please enlighten me!) This way, when you run your program it will function correctly.

Of course, always having to open the terminal to run IDLE3 is a bit of a

nuisance, so here's a handy shortcut: Instead of opening the terminal, just press alt-F2. A small window will open (see **Figure 4**) in which you can type the `sudo idle3` command and press enter or click the OK button. In a few seconds, IDLE3 will open, and you will have super-user privileges so that you can work with the GPIO pins.

That's all we have room for this month. Next time, we'll continue our GPIO output experiments by implementing a Cylon Eye project, or perhaps I should say Pylon Eye. (If you have a 10-segment LED bar display on hand, you may want to try that one on your own.)

After that, we'll move onto experimenting with GPIO inputs which will provide us with a convenient method of escaping from an infinite loop. If I don't talk too much along the way (fat chance!), we should also be able to get started with the basics of PICAXE-Pi communications.

In the meantime, have fun! **NV**

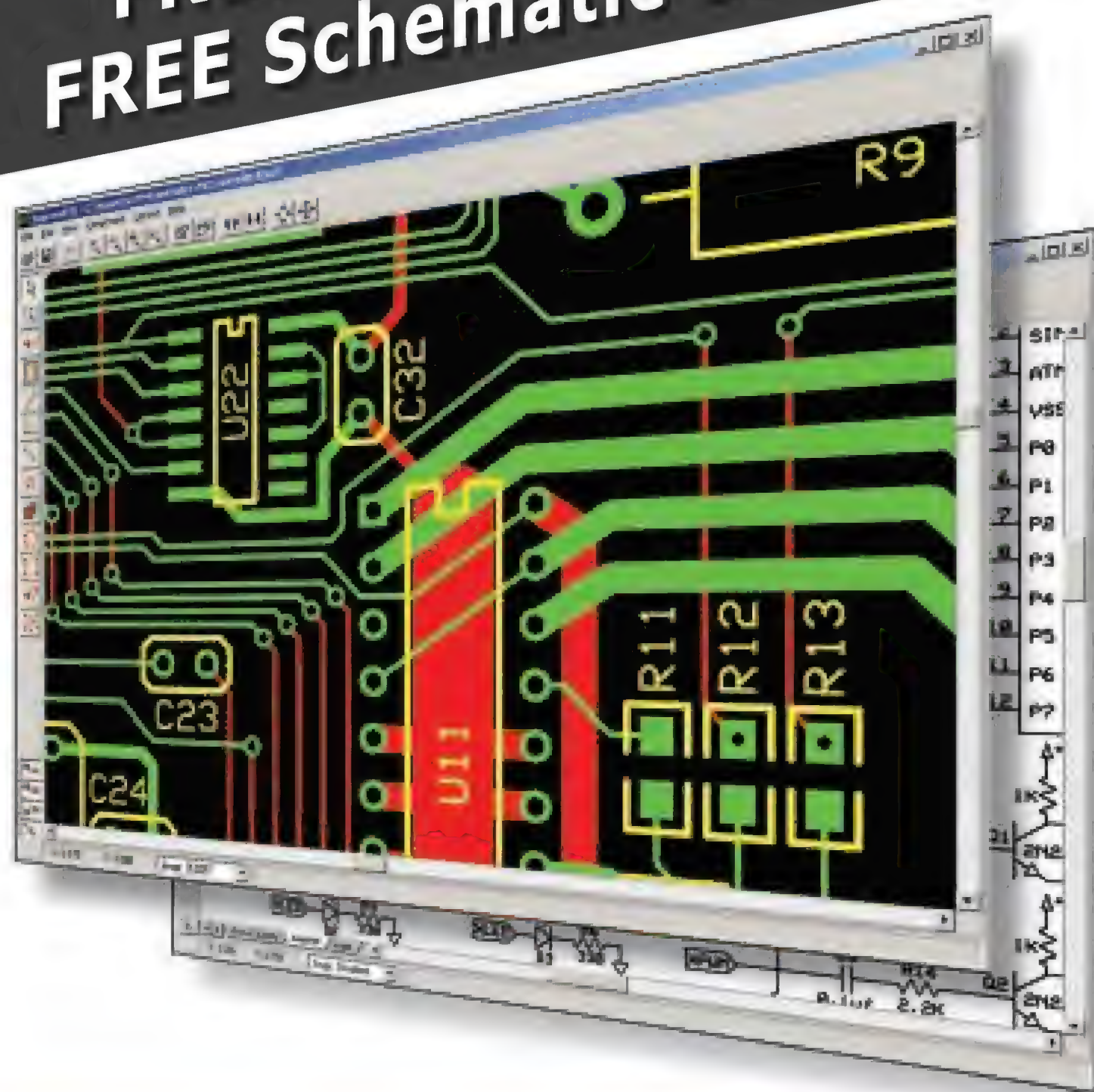
Books:

- *Programming the Raspberry Pi*, Simon Monk, 2012.
A brief introduction to the RPi and Python 3.1 programming – good first RPi book.
- *The Quick Python Book (Second Edition)*, Naomi R. Ceder, 2010.
A thorough introduction to Python 3.1.

Web Resources:

- The Raspberry Pi Foundation:
www.raspberrypi.org
- The Official Python Website:
www.python.org
- Python 3 Tutorial:
<http://docs.python.org/3.2/tutorial/index.html>
- Python 3.2 Standard Library:
<http://docs.python.org/3.2/library>
- Raspberry Pi Wiki/Hub – A comprehensive collection of helpful info:
http://elinux.org/RPi_Hub
- *The MagPi* – a monthly on-line RPi magazine (downloadable PDFs):
www.themagpi.com
- The Adafruit "Learn Raspberry Pi" Tutorials:
<http://learn.adafruit.com/category/learn-raspberry-pi>
- Think Python – Free, downloadable PDF version of the printed book:
www.greenteapress.com/thinkpython/
- Comprehensive Python 2 Tutorial:
<http://learnpythonthehardway.org/book>
- Another Comprehensive Python 2 Tutorial:
www.tutorialspoint.com/python/index.htm
- Good Beginning Tutorial for Python 3:
www.jeremymorgan.com/tutorials/raspberry-pi/raspberry-pi-programming-python/

\$51^{For 3} PCBs
FREE Layout Software!
FREE Schematic Software!



- 01 DOWNLOAD our free CAD software
- 02 DESIGN your two or four layer PC board
- 03 SEND us your design with just a click
- 04 RECEIVE top quality boards in just days

expresspcb.com

Finding the right parts for your design can be difficult, but you also don't want to spend all your time reinventing the wheel (or motor controller). That's where we come in: Pololu has the unique products — from actuators to wireless modules — that can help you take your design from idea to reality.



Sub-Micro Servo

ITEM #1053

\$4⁹⁵

- 3.7 g
- Specs at 6 V:
 - 6 oz-in
 - 0.07 sec/60°

Other servos available



Micro Metal Gearmotors

\$15⁹⁵

Other sizes available

- Dozens of motor winding and gear ratio combinations stocked
- Dual-shaft versions available for mounting encoders



Mini Maestro 12-Channel USB Servo Controller

ITEM #1352

\$29⁹⁵

- USB, serial, and internal scripting control
- 6-, 18-, and 24-channel versions also available



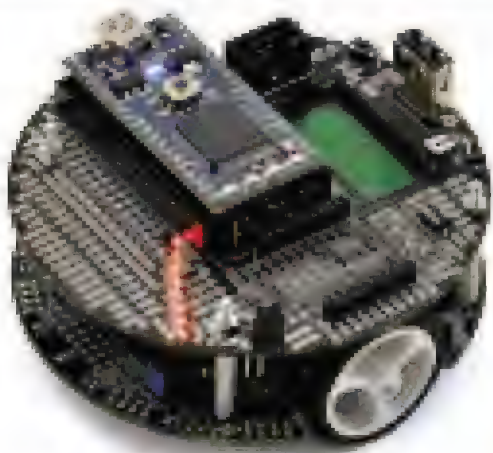
Wixel USB Wireless Module

ITEM #1337

\$19⁹⁵

Arduino shield available

Programmable MCU module featuring a 2.4 GHz radio and USB: write your own software or load precompiled, open-source apps.



m3pi Robot

ITEM #2153

\$189⁹⁵

3pi upgrade kit available

Enhanced version of our popular 3pi robot. Uses the 32-bit mbed development board to offer more processing power and free I/O lines.



Addressable RGB 30-LED Strip, 1 m

ITEM #2540

\$34⁹⁵

Waterproof, individually addressable RGB LED strip that runs on 5 V. Can be chained to form longer strips or cut for shorter sections. Also available in 2 m and 5 m lengths.



Pololu Basic 2-Channel SPDT Relay Carrier with 12 VDC Relays

ITEM #2487

\$8²⁵

Control two SPDT relays easily with logic level signals. Available with 5 V relays or without relays. Single carrier also available.



Wall Power Adapters

STARTING AT

\$5⁹⁵

- Center positive, 5.5 mm x 2.1 mm barrel connector
- Available voltages: 5 V, 9 V, and 12 V
- Available currents: 1 A, 3 A, and 5 A



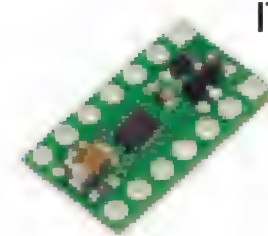
170-Point Breadboard

ITEM #1491

\$2⁹⁵

Other colors available

- 17 rows of tie points for up to two 14- or 16-pin DIP ICs
- Tabs allow multiple units to be connected for larger projects



ITEM #2135

\$4⁴⁹

- DRV8835 driver
- 2 V to 11 V operating range
- 1.2 A continuous (1.5 A peak) per channel
- 14-pin DIP form factor



ITEM #713

\$4⁹⁵

- TB6612FNG driver
- 4.5 V to 13.5 V operating range
- 1 A continuous (3 A peak) per channel



Dual Motor Drivers

ITEM #2130

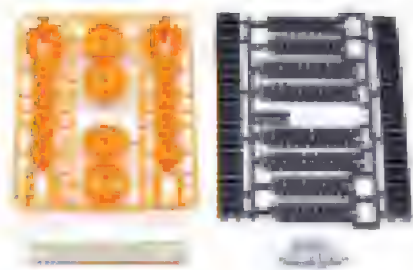
\$4⁹⁵

- DRV8833 driver
- 2.7 V to 10.8 V operating range
- 1.2 A continuous (2 A peak) per channel

Tamiya 70097 Twin-Motor Gearbox Kit

ITEM #61

\$8⁹⁵



Tamiya 70100 Track and Wheel Set

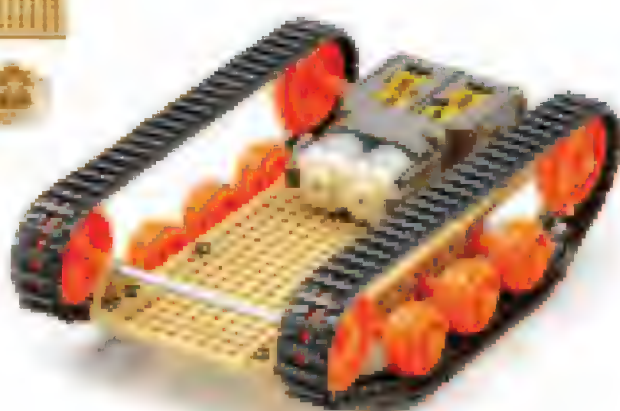
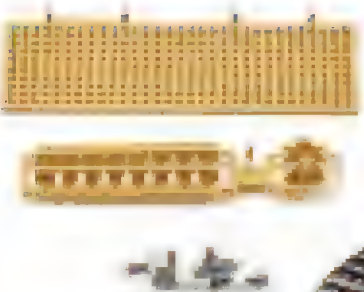
ITEM #106

\$7⁹⁵

Tamiya 70098 Universal Plate Set

ITEM #79

\$6²⁵



Our large selection of Tamiya gearboxes and construction sets is great for building inexpensive mechanisms and robot chassis!



Crimping Tool

ITEM #1928

\$34⁹⁵



26 AWG Red Stranded Wire, 70 ft

ITEM #2622

\$4⁵⁰

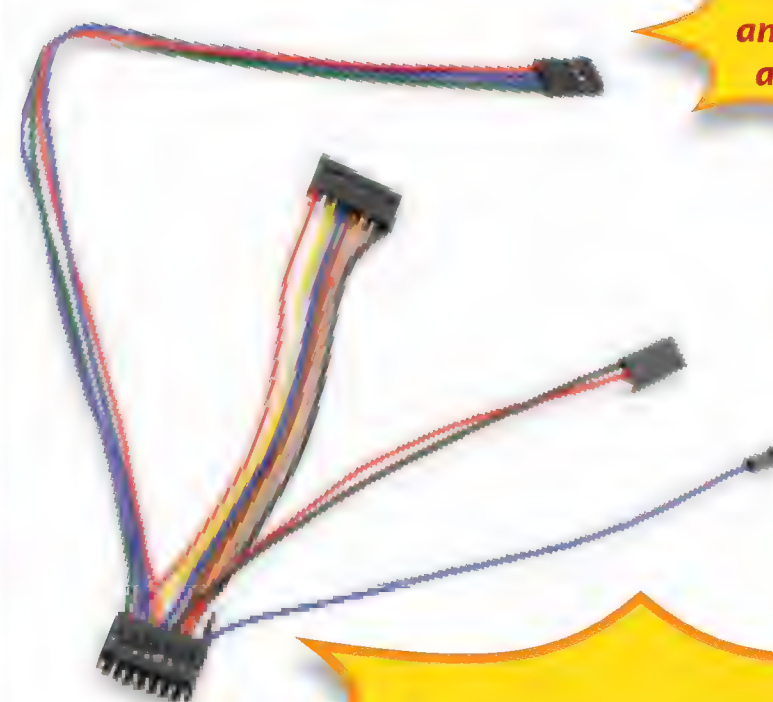
Other colors and gauges available



Crimp Connector Housings

STARTING AT

59¢/pack



Female Crimp Pins

ITEM #1930

\$5⁹⁵/100-pack



Male Crimp Pins

ITEM #1931

\$7⁹⁵/100-pack

16-26 AWG Wire Stripper

ITEM #161

\$6⁹⁵

Everything you need to make custom cables!

In this column, Russ answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. Send all questions and comments to: Q&A@nutsvolts.com

- Tesla Coil Project
- Backup Camera Mod
- Mini LTspice Tutorial

Go to www.nutsvolts.com/index.php?/magazine/article/october2013_Q&A to post comments on these topics.

InfraRed Pulse Amplifier

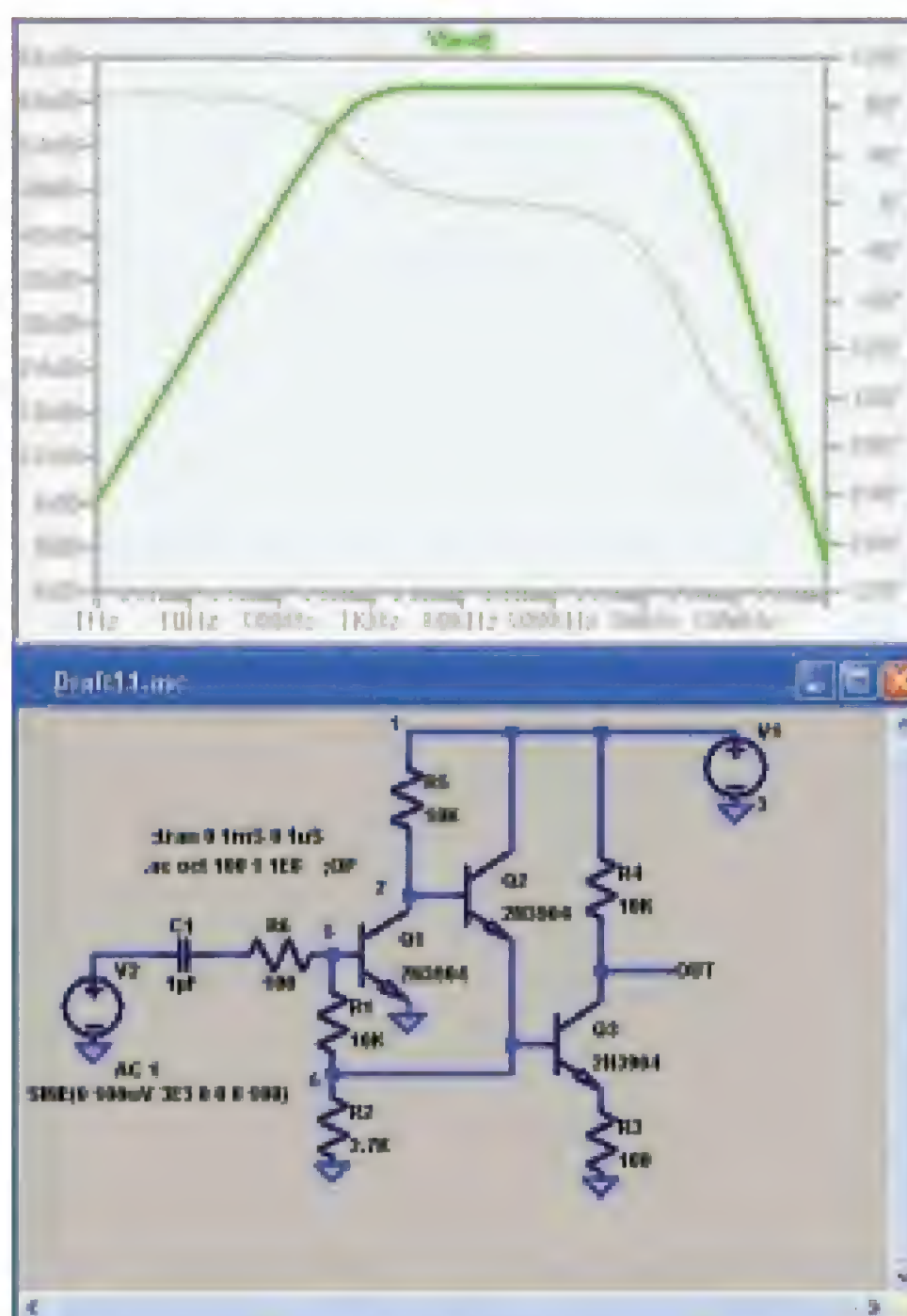
Q I have a fruit garden that is about 300 yd x 300 yd. I designed a special pulsed laser diode and an IR receiver (with a few mirrors around corners), and created an invisible beam around my garden. If the beam breaks, I know there is an intruder.

In order to do this, I designed an encoder for the transmitter side that puts out 3.0 kHz (pulse width = 300 nSec per pulse) and a decoder for the receiver side to verify that I'm receiving the right beam. The first amplifier I designed (three years ago) was consuming over 20 mA. I tried to simplify the design and used better circuitry to reduce power consumption. My second design which consumes about 1.5-2 mA is shown in **Figure 1**. The received signal is not as clean as it was on my first design, but it does work.

I'm wondering if there is a way to bring power consumption below 0.5 mA and not lose quality of signal out?

— Sam Jannati

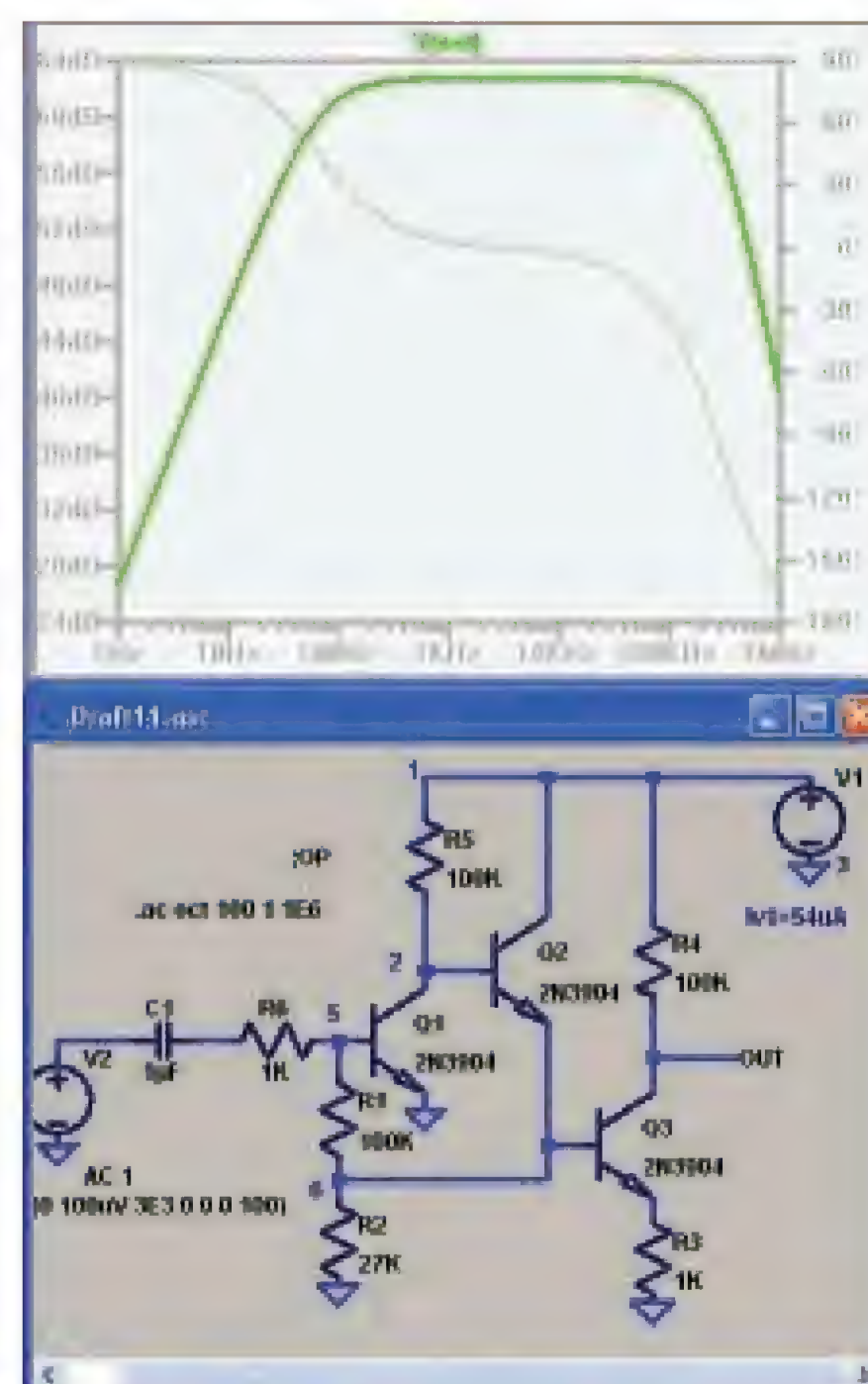
A Your design in **Figure 1** is excellent; the input impedance is low, but the bandwidth is almost 10 MHz. I changed C1 to 1 μ F because the low frequency roll-off was above 3 kHz with 0.1 μ F. My simulation in LTspiceIV shows the supply current to be 0.54 mA, so you need a lower power comparator. The TLV3401



■ FIGURE 1.

runs on 3V, and only draws 550 nA plus the pullup resistor current. The TLV3401 response time is measured in microseconds (5 μ S fall time), so you will need to widen the pulse width. Neither the rise time nor the output capacitance is specified, so I expect it is pretty bad. Even so, with a 333 μ S period, a usable pulse width should be obtainable.

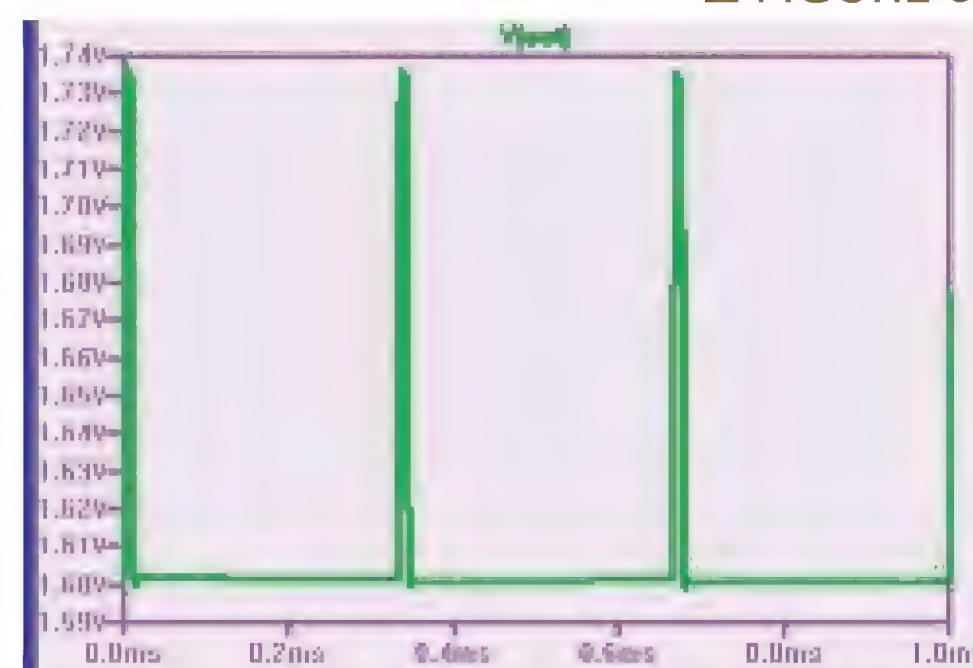
You can further reduce the power at the expense of bandwidth by increasing the resistor values. I increased them all by 10 and got the supply current down to 54 μ A. The frequency response is still decent; check it out in **Figure 2**. I drove the circuit of **Figure 2**



■ FIGURE 2.

with a 100 μ V 10 μ S pulse with the result shown in **Figure 3**. The comparator should respond to a 10 μ S pulse with a 10K resistor. If the output is a negative pulse, the current will be $(10/333) \times 3/10K = 9 \mu$ A average.

■ FIGURE 3.



Tesla Coil Project

Q Do you have a schematic for building a traditional Tesla coil and a solid-state Tesla coil using a TV flyback transformer? I would like to build both of them.

I have always enjoyed your Q&A answers.

— Brian J. Miller

A By traditional, I think you mean a coil you can build yourself. Homemade Tesla coils typically involve a paper or plastic tube, two or three inches in diameter, and fairly long. I wanted to simulate the coil in LTspice, so I used a formula from the ITT handbook for solenoid coils that is said to be 1% accurate:

$$L = N^2 * (R^2 / (9 * R + 10 * C))$$

where

L is in microhenries

R is the coil radius

N is the number of turns

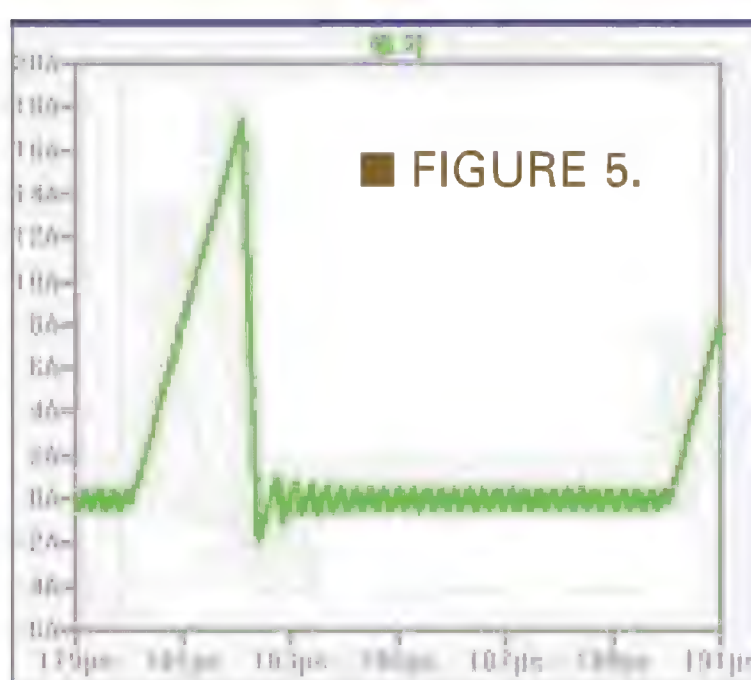
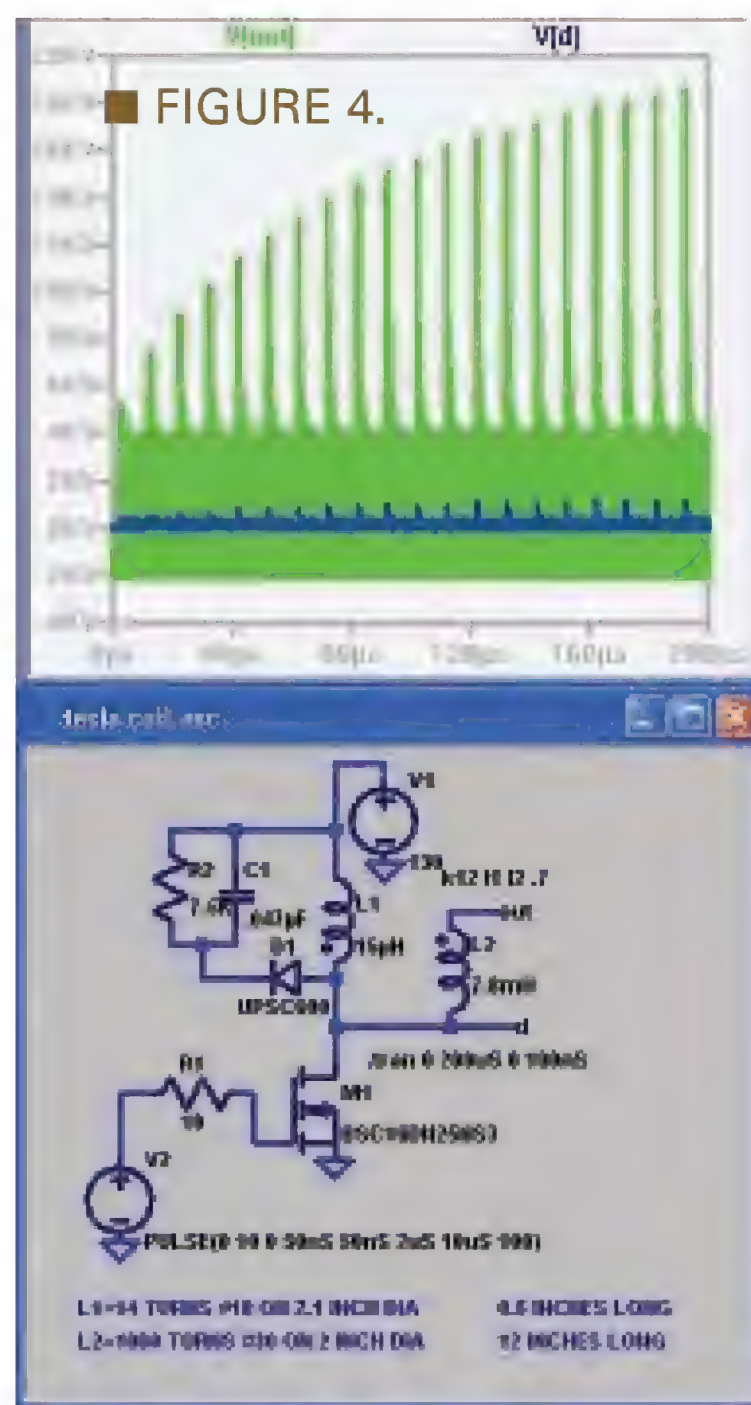
C is the length of the coil

(L was already used)

The inductance of 1,000 turns on a two inch diameter form is 7,827 microhenries (7.8 mH). Using that as the secondary, I adjusted the primary inductance to get 20 KV output, then calculated the primary turns. I set the coupling factor at 0.7, thinking that with an air core and different lengths of coils the coupling would not be very good. That was a guess.

Since the voltage depends on the rate of change of current, it is essential to have a fast switching transistor driving the primary. In **Figure 4**, when the transistor turns on the current builds up linearly until the transistor turns off (see **Figure 5**). When the transistor turns off, the current continues to flow, charging the stray capacitance and driving the voltage to a very high value.

The transistor used here (IXBP



5N160 G) has a voltage rating of 1,600 volts; you do not want to exceed that. ICs and transistors run on smoke, you know; if the smoke gets out, it won't work.

The capacitor C1 and resistor R2 function to limit the peak voltage. The capacitor is charged through D1 by the flyback and is discharged by R2. At some point, the charge and discharge energy becomes equal and the charge on the capacitor stabilizes. In this case, the stable voltage is 1,200 volts. The power in the resistor is 150 watts, and the diode needs to be rated 2,000 volts (you could series three 600V diodes from the same batch).

Peak current in the diode is 18 amps, but the duty factor is 1/20 so a one amp diode should work. Use a

high speed diode; a 1N4000 type won't turn off fast enough.

The saturation voltage of the 5N160 is about six volts and the peak current is 18 amps, but with the duty factor of 3.5/10, the average is 3.15 amps which makes the power dissipation of the transistor 20 watts. With a good heatsink, the TO-220AB package should handle that or the TO-247 package could be used.

It appears that the fall time of the current is 0.4 microseconds, therefore the voltage should be:

$$E = L * di/dt = 15 \mu H * 18 A / 0.4 \mu s = 675 V$$

I can't explain why the voltage on C1 is approximately double that. You may find when you build it that the output is half what was expected and the C1, R2 protection is not needed (so much for simulation!).

The circuit using the TV flyback transformer will be the same. You will have to experiment with the R2 and C1 values so as not to blow up the transistor.

The circuit of **Figure 6** is a pulse generator to drive the 5N160; it is intended to be a guide. I have not built any of it. Q1, Q2, and Q3 are an op-amp with positive feedback through R5 to charge and discharge C1.

There is negative feedback through R7 to provide hysteresis. When the voltage on C1 is higher than the voltage on Q2 base, the output goes low and discharges C1. When the voltage on Q1 base is lower than the voltage on Q2 base, the output goes high and the cycle is complete.

Backup Camera Modification

Q My 2013 KIA Optima has a backup camera and only works when I place it in reverse. I would like to be

able to turn it on and off while driving to see tailgaters. My guess is to put an on/off switch between the reverse switch, but I have no schematic. Can this be done?

— Walt Krol

A The KIA shift lever is in the center console. Most likely the switch is connected to it and you will have to remove the console cover; a repair manual will be handy for doing that. The switch is normally open, so all you need to do is connect another switch in parallel so it becomes an either-or circuit.

There is no need to cut any wires; you can get a splice at the hardware store.

The splice is a plastic part with metal inserts that cut the insulation but not the wire. There is a channel for the through wire and a short channel for the splice. Place the wires in the channels, fold the top over, compress it with pliers, and snap the retainer over the edge. Done.

It is possible that the switch is inside the transmission, but I doubt it. The repair manual should tell you, but in any case, the modification is the same.

It is likely that the switch also turns on the backup lights. In most states, it is illegal to have white lights at the rear of the vehicle while it is going forward, so don't drive around with the backup lights on.

Newbie Questions

Q I am a new subscriber to the magazine and am a beginner when it comes to electronics. I got the magazine because I wanted to practice reading schematics and datasheets.

I read the book, *Make: Electronics* by Charles Platt as an introduction to electronics but I am still struggling with schematics.

I wanted to try to understand the simple cat alarm schematic proposed in the August 2013 Q&A section.

I'm sure these questions are really basic, so thank you for helping me.

LTspice Tutorial

My LTspiceIV is a free simulation program that has been enhanced for switching circuits (SPICE typically has a problem with fast switching circuits; you'll get a "time step too small" error message).

The program has schematic capture and a waveform viewer. You can download it from www.linear.com/designtools/software. The manual is available in the HELP file in the upper right corner.

There is an extensive library of Linear Technology parts, but if you want to use some other device or a transistor that is not in the library, you will have to find the SPICE file and install it. To do that, use a search engine (Google) and when the file is found, copy and paste it into Notepad.

Now, go to C:\program files\LTC\LTspiceIV\lib\sub and copy the file to that subdirectory if it is a subcircuit. Now you need a symbol, so go back to \lib\sym and double-click on the type of circuit, for example, an op-amp. The LT1012 is a typical op-amp with input, output, and power pins that will fit many applications.

If you double-click on the symbol, it will open and you can change the name. If you right-click on the red circle at the name, a window opens that will allow you to change the name. If you left-click, then you can choose delete. If you delete, then use Draw/text to make a new name. Save the symbol with a new name which must be the same as the name of the subcircuit. You are not done yet.

The order of the pins in the symbol must be the same as the order of pins in the subcircuit. For a five-pin op-amp, the order is:

1. Positive input
2. Negative input
3. Positive power
4. Negative power
5. Output

The actual numbers don't matter; only the order is important. If the order is different, edit the subcircuit.

LTspice will make the symbol for you. In the SPICE file, right-click on the line that has the name of the subcircuit and choose "Create Symbol" from the pop-up window.

If you have a transistor .model file, go to C:\program files\LTC\LTspiceIV\lib\cmp and double-click on standard.bjt. If it doesn't open as a text file, open it with Notepad. Just copy and paste the file and you are done.

If you want the library to display the manufacturer, Vceo rating, and Ic rating, you can add that at the end of the file; just use a space as a delimiter. The syntax is: Vceo=xx icrating=xx mfg=yyy.

If there is not a subcircuit model available, you may be able to make one using LTspice. Just draw the schematic using simple parts. If it is too detailed, the simulation may take a long time to run. Instead of a transistor, use a current controlled current source; instead of an op-amp, use a voltage controlled voltage source because these will run much faster.

For example, I used an INA128 instrumentation op-amp recently (I don't know if a model is available; I am going to make one as an exercise). The IC has three op-amps and some resistors.

To start, I will simulate the over-voltage protection

1) Is Figure 3 only for the transmitter? So, would a receiver near the door be required in addition to this?

2) Is the 555 timer used to turn the transmitter on and off to minimize the current it draws?

3) The schematic for the 555 timer is a little different from the ones in my book. One thing that confuses me is what is going on with the CNTRL. It looks like it goes into R2. It looks almost like a potentiometer but without any arrow.

4) What is IC2P?

Thanks so much and sorry again for the newbie questions.

— Andrew Martin



You came to the right place, Andrew. I love to answer questions so feel free to ask them any time.

1.) A communications system requires both a transmitter and a receiver. The distance that it will work over depends on the transmitter power, the efficiency of the antenna, and the sensitivity of the receiver. In this case, the transmitter is low power (under 100 milliwatts) and the distance is short, so an efficient antenna is not required. I don't know what the sensitivity of the receiver is, but it is usually measured in microvolts.

2.) Yes, power = volts x amps. So, if the voltage is zero part of the time,

the power is reduced. The shorter the on time, the lower the power. Note that the 555 is LMC555; this means it is CMOS and low power.

The CMOS 555 runs on three volts, but the bipolar 555 (LM555) draws more current and doesn't work below 5V.

3.) My bad, I put the resistor too close to the 555. Actually, the CNTRL pin is not connected to anything.

4.) IC2P is a mistake; I must have had an IC2 at one time and forgot to remove the power pins. In my schematic software, the power pins can be moved so as to have an uncluttered schematic.

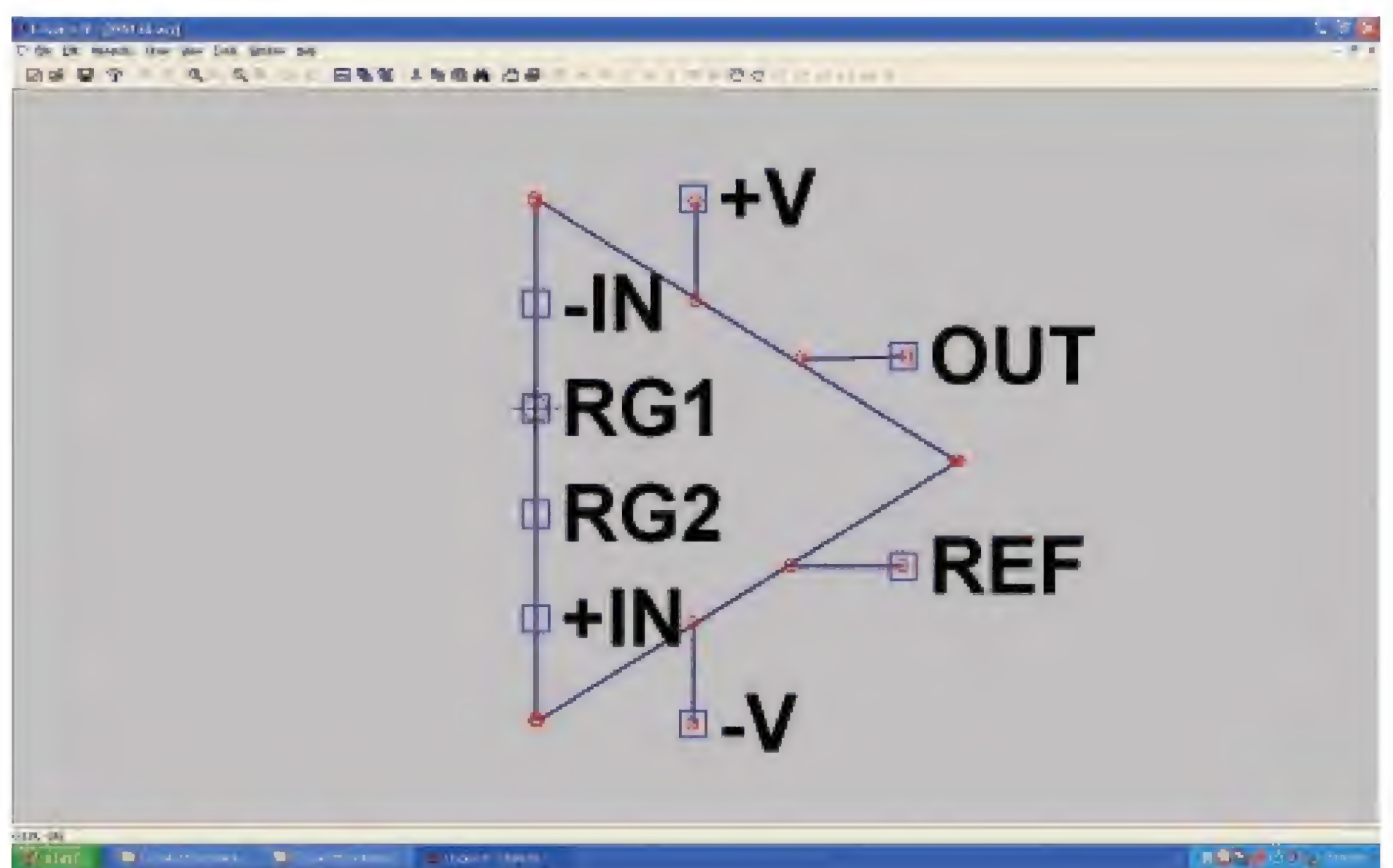
I hope this helps. If not, just let me know.

with two diodes and a resistor (R1, D1, D2; see **Figure 6**). I don't know the value of R1, but it doesn't matter because the input resistance of E1 is infinite until the diodes conduct. The current in that condition is a minor consideration. The gain stage A1 is simulated by a voltage controlled voltage source E1. Before placing E1, I mirrored it by clicking on the E3 symbol at the upper right, then used ctrl-r twice to rotate it.

The offset voltage is simulated by a series voltage source V1 and V2. I probably could have used just one voltage source, but used two for symmetry. Note that V2 is rotated so that the two offsets add and do not cancel. The bias current is simulated with a shunt current source. It can't be in series because the infinite resistance of E1 will try to produce an infinite voltage. The offset voltage produced by the bias current is limited by the impedance of the source.

I want the terminal node numbers to be the same as the pin numbers of the IC. To do that, right-click on the wire and select "label net" from the pop-up window.

E1 has to have a high gain; to set the gain, right-click on the symbol and double-click on the "E" of value. Enter the desired value; in this case, 1E7 (10000000). E1 has infinite bandwidth but I want the model to be band limited — similar to the actual device.



■ FIGURE 6.

C1 and C2 are intended to do that; the initial value was set at 1 PF, then adjusted by fiddle and fudge to match the IC response. I might need capacitors across R7 and R8, but I'll wait and see.

My first attempt with only E1 to simulate A1 had constant bandwidth regardless of gain. The typical op-amp response is a constant gain-bandwidth product; to simulate that, I figured that slew rate limiting was needed. To simulate that, I used R9, C2, E4, and lowered the gain of E1. The gain of E4 changes the bandwidth.

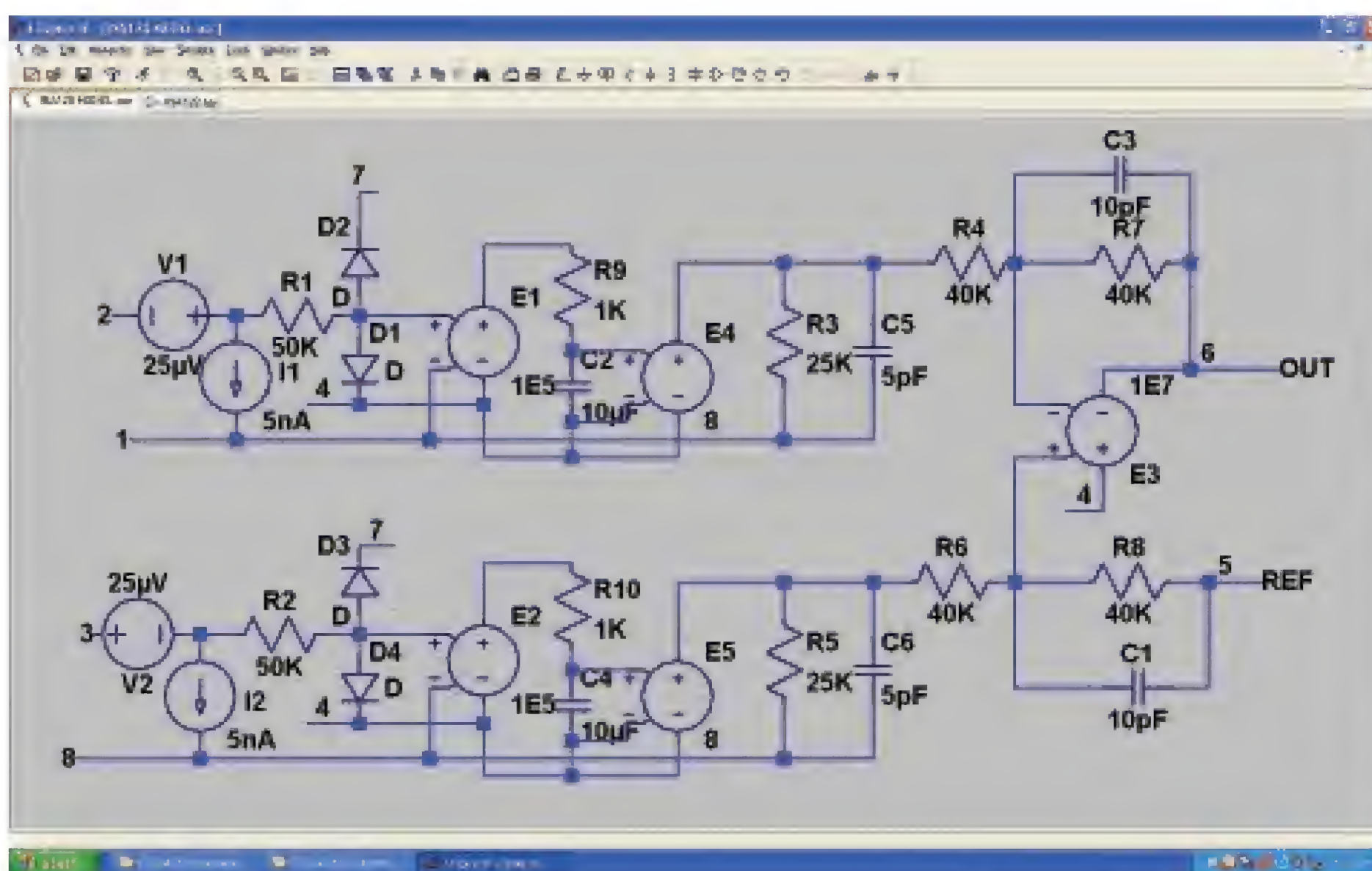
The result is close to 1 MHz bandwidth at unity gain and 20 kHz bandwidth at 60 dB gain; close enough

for my purpose.

This completes the model; now use VIEW/SPICE NETLIST and copy and paste to Notepad. I could make a symbol and call it INA128.asc, then LTspice would use it like a subcircuit, but that is not a portable model.

In Notepad, insert the .SUBCKT command at the top of the file; the format is: .SUBCKT name node1 node2, etc. At the bottom of the file, put .ENDS or .ENDS INA128. Now, save the file with the name INA128.SUB in C:\programfiles\LTC\LTspiceIV\lib\sub. The automatic symbol creation did not work for me, so I drew it myself (file/make symbol; see **Figure 7**). Just be sure that the order of pins is the same as the order of nodes in the subcircuit call.

That's a wrap for this time. **NV**



■ FIGURE 7.

MAILBAG

Re: Ampere-Hour Meter, August 2013, page 22:

Mr. Kincaid, your schematic indicates the unused LCD data pins are floating. This could be the problem instead of the software. I noticed in another article in the same August issue that a similar LCD had its unused data pins tied to ground. I hope this helps.

Ray Malcuit

Thanks Ray, I should have known that.

Re: Ampere-Hour Meter, August 2013, page 22:

Hi Russell! You were asking for help in the latest issue (just got it today); not sure if you've fixed it, but I can help you a bit.

As far as R/W goes, GROUND IT like the diagram shows. You only have to look at that pin if you are doing an LCDIN command; in other words, reading stuff from the memory on the display. Using only LCDOUT, grounded is fine. Hook it up just like the diagram.

You also haven't set up the chip properly. I looked at the datasheet briefly, and the 16F883 does have an ANSEL register on some of its ports. If you don't tell the port to be digital (which you want for all ports except the one you are reading the analog voltage from), it defaults to analog and won't work properly.

On the LCDOUT command, you are using a # symbol before the variable that you are sending to the LCD. Not sure where the # symbol came from, but I don't think

that's going to work. From your description, you want to display the amperage in decimal, so we have to tell it to send the ASCII characters for decimal digits; like this:

```
LCDOUT $FE, 1, "AMPERES = ", DEC3 AMPS
```

This would display the amps as a three-digit decimal number, with leading zeros showing. DEC4 would be four-digit, etc. Look at MODIFIERS in the PBP manual.

I have included a section of code from a project of mine (a temperature controller for my smoker) that uses two analog inputs (the thermocouples), an LCD, and various other digital ports. This is for a 16F886, but the principles remain the same.

The last few lines are some of the LCDOUT commands that I use (they work) — not in the proper place, just randomly placed.

Remember that comments are preceded with a single quote.

Hope this helped.

Andy

Thanks very much for you analysis. I think I can make it work now.

```
*****Start code snippet*****
```

```
#CONFIG
__config __CONFIG1, __INTRC_OSC_NOCLKOUT &
__WDT_OFF & __MCLRE_OFF & __LVP_OFF & __CP_OFF
```



```
#ENDCONFIG
```

```
DEFINE OSC 8 'LETS PBP KNOW THE OSCILLATOR IS  
RUNNING AT 8MHz
```

```
DEFINE NO_CLEARWDT 1 'NO WATCHDOG TIMER - FOR  
NOW - WILL DO OUR OWN CLRWDT
```

```
OSCCON = %01110001 '8 MHz INTERNAL OSCILLATOR,  
INTERNAL OSCILLATOR IS USED FOR SYSTEM CLOCK
```

```
DISABLE DEBUG'NO PBP INTERRUPTS NO PBP DEBUG
```

```
' Set LCD Data port
```

```
DEFINE LCD_DREG PORTA
```

```
' Set starting Data bit (0 or 4) if 4-bit bus
```

```
DEFINE LCD_DBIT 4
```

```
' Set LCD Register Select port
```

```
DEFINE LCD_RSREG PORTC
```

```
' Set LCD Register Select bit
```

```
DEFINE LCD_RSBIT 4
```

```
' Set LCD Enable port
```

```
DEFINE LCD_EREG PORTC
```

```
' Set LCD Enable bit
```

```
DEFINE LCD_EBIT 5
```

```
' Set LCD bus size (4 or 8 bits)
```

```
DEFINE LCD_BITS 4
```

```
' Set number of lines on LCD
```

```
DEFINE LCD_LINES 2
```

```
' Set command delay time in us
```

```
DEFINE LCD_COMMANDUS 1500
```

```
' Set data delay time in us
```

```
DEFINE LCD_DATAUS 44
```

```
DEFINE ADC_BITS 10 ' Set number of bits in  
result
```

```
DEFINE ADC_SAMPLEUS 50 ' Set sampling time in  
uS
```

```
ADCON0 = %11000000 ' Set ADC_CLOCK to Fosc/32,
```

```
ADCON1 = %10000000 ' Right-Justify result in
```

```
ADRESH:ADRESL registers, use Vss and Vdd as
```

```
voltage references
```

```
ANSEL = %00000011
```

```
' Set AN2-AN7 to digital, AN0-AN1 to analog
```

```
ANSELH = %00000000
```

```
' Set AN8 and higher channels to digital
```

```
' operation
```

```
TRISA = %00000011
```

```
' Set PORTA.0 & 1 input, the rest are outputs
```

```
TRISB = %00011111
```

```
'B0-4 INPUTS, ALL THE OTHERS OUTPUTS
```

```
TRISC = 0
```

```
'PORTC.0 ALL OUTPUTS
```

```
OPTION_REG = 0
```

```
'PORT B PULLUPS ENABLED, TMR0 PRESCALER IS
```

```
*NOT*
```

```
'ASSIGNED TO WDT
```

```
WPUB = %00011111
```

```
'PULL UPS ENABLED ON PORTB.0-4
```

Superior Quality Soldering Tools from

HAKKO

NEW



FX-951

HIGH PERFORMANCE SOLDERING STATION

- Digital temperature display
- Superior heat transfer
- Fast thermal recovery
- 400°- 840°F (200°- 450°C)
- Low temperature alarm
- Sleep mode for longer tip life



FX-601

TEMP-ADJUSTABLE SOLDERING IRON

- Electronic "closed loop" control
- 464°-1004°F (240°- 540°C)
- Ceramic heating element
- Lightweight (2.4oz w/o cord)
- Extra long 3-plug, 6 ft.cord
- UL/CSA Approved



FX-888D

LOW COST, DIGITAL SOLDERING STATION

- Digital display (°F or °C)
- Adjustable temperature
- 120°-899°F (50°- 480°C)
- Idle temp within 1.8°F (1°C)
- Ergonomic, handpiece
- Ceramic heater

For complete information, visit HakkoUSA.com/NutsVolts

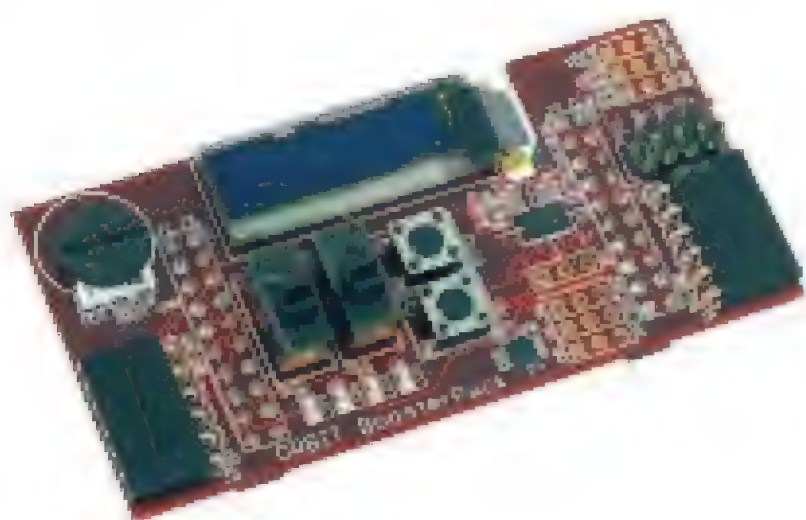


NEW PRODUCTS

■ HARDWARE
■ SOFTWARE
■ GADGETS
■ TOOLS

I/O EXPANSION BOARD FOR TI STELLARIS LAUNCHPAD

Digilent has announced the release of the Orbit BoosterPack™ — an add-on board for the Texas Instruments Stellaris LaunchPad microcontroller evaluation kit. The Orbit greatly expands the



input/output capabilities of the LaunchPad, as well as introducing Digilent Pmod expansion connectors.

The Orbit BoosterPack includes two 1x6 Digilent Pmod connectors, a three-axis accelerometer, a 256 Kbit I²C EEPROM and I²C temperature sensor, a 128x32 pixel OLED display, an analog potentiometer, as well as other features.

The Stellaris LM4F120 LaunchPad evaluation board is a low cost evaluation platform for ARM Cortex-M4F-based microcontrollers from TI. The design of the LaunchPad highlights the LM4F120H5QR microcontroller with a USB 2.0 device interface and hibernation module. With the TI Launchpad and Orbit Booster, users can learn how to program the ARM microcontroller by visualizing the result.

A collection of tutorials targeting the use of the TI Launchpad and Orbit Booster Pack has been created by several educators, including Craig Kief, deputy director of COSMIAC — a space technology research center at the University of New Mexico.

These tutorials provide an introduction to microcontrollers, with the topics ranging from beginning material — like loading the software — to more advanced topics — like accelerometers and sensors. They are now used in microcontroller workshops funded by the National Science Foundation (NSF) and are available to download for free at the COSMIAC Microcontroller website

Digilent's Orbit Booster Pack and TI Launchpad are available for \$29.99 and \$14, respectively.

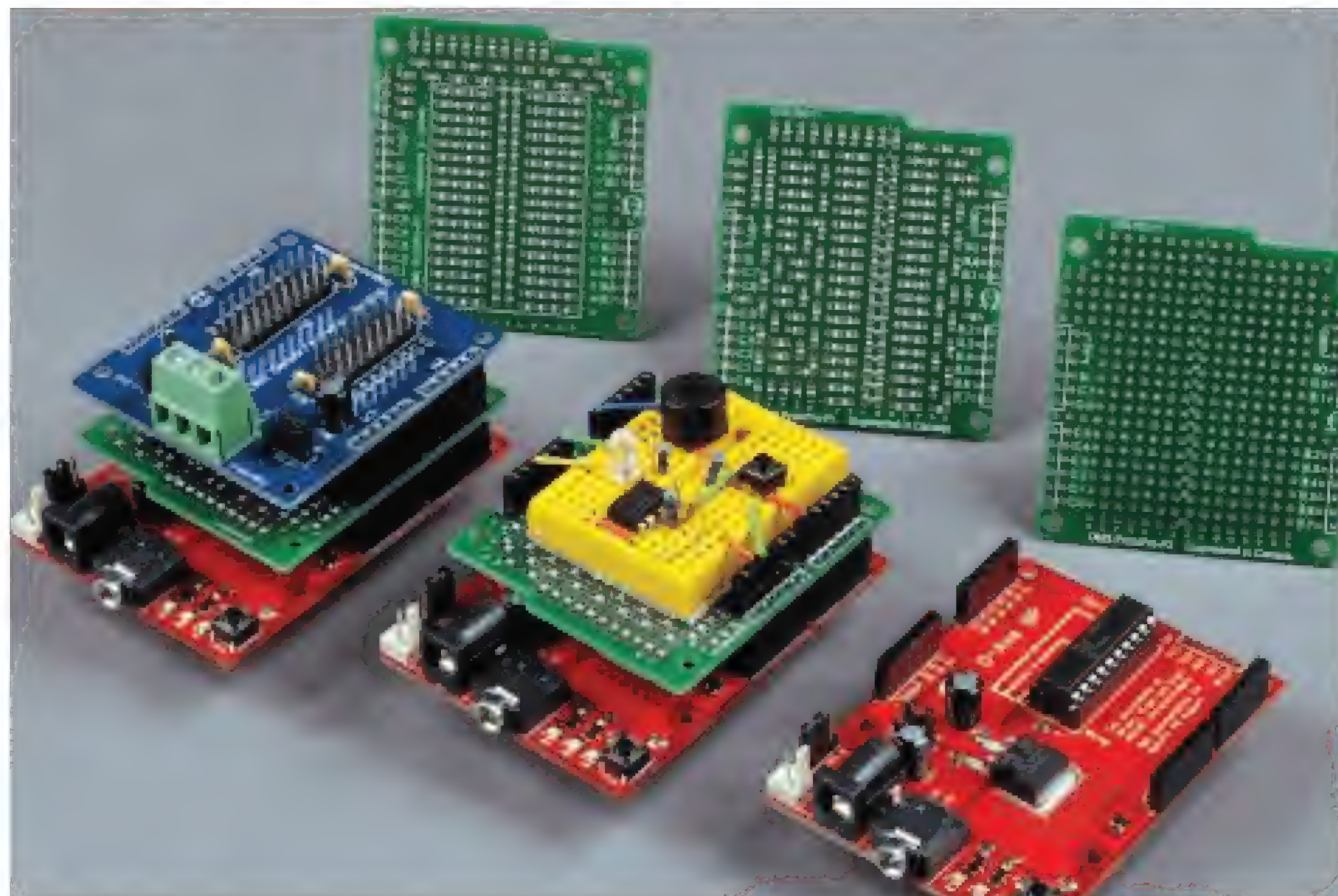
For more information, contact:
Digilent, Inc.
Web: www.digilentinc.com

PROTOTYPING SYSTEM FOR PICAXE MICRO CONTROLLERS

Aztec MCU Prototyping now offers the D-Axe BP prototyping system intended primarily for use with PICAXE microcontrollers. Engineered in Canada and carrying their house brand — Omega MCU Systems (OMS) — this new system provides the user with a simplified yet versatile, expandable, and modular platform from which to develop prototypes based on PICAXE eight-, 14- and 20-pin microcontrollers. As with all of the OMS branded products, it is designed to speed up the building of prototypes and make the whole process more reliable and repeatable.

The D-Axe BP system is based on the D-Axe BP main board and accepts modular BakPak plug-in boards for expansion. All signals from the microcontroller are brought out to the female headers into which the expansion boards plug. This allows the builder to add functionality one board at a time to achieve their project goals.

The D-Axe BP can be connected to solderless breadboards using



Recycling & Remarketing High Technology

WEIRDSTUFF[®]

WAREHOUSE

Software, Computers, Electronics, Equipment, Doodadicks

WE BUY/SELL EXCESS & OBSOLETE INVENTORIES

FREE COMPUTER AND ELECTRONIC RECYCLING

GIANT 10,000 SQ. FT. AS-IS SECTION

384 W. Caribbean Dr.
Sunnyvale, CA 94089
 Mon-Sat: 9:30-6:00 Sun: 11:00-5:00
(408)743-5650 Store x324
WWW.WEIRDSTUFF.COM

www.Primecell.com

Battery rebuilding service

Dead Batteries? Don't toss them. Send them to us - our rebuilds are better than original specifications.

Tools

Electronics

Radios

Visit www.primecell.com for important details
 24 Hr Secure recorder tel-fax (814) 623 7000
 Quotes email: info@primecell.com
 Cunard Assoc. Inc. 9343 US RT 220 Bedford PA 15522

d6 LABS

RFM22B
\$6.95 ea 100mW Transceiver

RFM50B
\$9.90 ea 100mW Transceiver w/8051 uC

www.d6labs.com

jumper wires, allowing it to integrate with any existing system. Manufactured on high quality 1.6 mm thick double-sided FR4 fiberglass printed circuit boards with 1 oz copper traces, this system is built for long life under hard use.

The main board features a 3.5 mm stereo phone jack for compatibility with existing AXE026 and AXE027 PICAXE programming cables, and onboard LED indicators to monitor programming and data traffic.

An onboard power regulator capable of supplying more than 500 mA ensures that it can support sizeable expansion, and a pushbutton hard reset is provided for user-friendly operation. It is fully compatible with all current PICAXE programming software.

While the D-Axe BP is primarily intended for use with a PICAXE, it also supports a host of small PIC microcontrollers via an optional ICSP header.

The initial release of the system includes the D-Axe BP main board (shown in red), a mini-breadboard BakPak kit for building quick experiments, a Sensor/Actuator BakPak partial kit (blue) for use with three-wire sensor and actuator 'bricks,' and three prototyping BakPaks (green) with different foil patterns — the P1, P2, and P3.

The P1 pattern is optimized for DIP packages; the P2 is a free form design with unconnected plated through holes and lands; and the P3 duplicates the mini breadboard

LEMONS INTERNATIONAL Tel: Free: (866) 345-5867 Email: sales@lemonsint.com

RF Design Group

Headed by an experienced former NASA engineer with credentials from government agencies and large well-known corporations, our design team provides RF design solutions for the following sectors:

- Industrial • Lighting Control • SCADA
- Military • Law Enforcement • Space
- Medical • Smart Grid Metering • Tracking
- Alternative Energy • Consumer

The Lemos International Design Team is prepared to work with your in-house engineers or support your Rf project from initial design to implementation.

"Making your RF ideas into profitable products."
www.lemonsint.com

QKITS.COM

12V USB Relay cards are based on the Arduino physical computing controllers.

Program them as stand-alone using open source Arduino IDE.

KTA-223 has 4 optically isolated digital inputs, 3 analog inputs, 8 relay outputs.

KTA-225 has 8 analog inputs, 8 relay outputs.

Call 1 888 464 5487 to order!

In the Nuts & Volts Webstore NOW!

NUTS AND VOLTS

NV BOOK SPECIALS

HACKING ELECTRONICS

309 Pages
Only \$23.95

Raspberry Pi Projects for the EVIL GENIUS

224 Pages
Only \$19.95

SAVE 20%!

To order call 800 783-4624 or visit: <http://store.nutsvolts.com>

Limited time offer

SLIDE MOUNT for MOBILE RADIOS

UNIVERSAL: FITS MOST 2 WAYS

Model ET-850 Universal Slide Mount. Standard mounting platform, specify radio make/model. Ends blind fumbling to release hidden fasteners. Easily transfer radios from vehicle to vehicle. Also use in the office, job trailer, marine too. Black powder coat finish. Volume prices.

(Ask about our maximum security Locking Radio Cage™ Mount)

E TIP, Inc., North Aurora, IL 60542 USA
 (800) 530-5064 FAX (630) 801-9569
www.etipinc.com | sales@etipinc.com

E TIP is a Veteran Owned Small Business

NKC electronics

MDE8051 Trainer by Digilent

NKCElectronics.com/MDE8051

Includes the MDE8051 training board, power supply, serial cable

Purchase Orders are accepted from Educational Institutions, US Government and Research Centers

Continued on page 76

BUILD THE WIRELESS SUPER DETECTOR

By Ron Newton
sjnewt@att.net

Go to www.nutsvolts.com/index.php?/magazine/article/october2013_newton for any additional files and/or downloads associated with this article. You can also post comments.



I started off this article with the intention of designing a "You Have Mail" transmitter and receiver. However, I discovered my circuit has many other applications, so I decided to design a board which would widen the scope to a large group of sensors that can be used with the transmitter.

The following projects have locations included on the board for the measurement of: temperature, vibration, light, sound, motion, normally open switch, normally closed switch, any varying resistor input, voltage input, mA input, tilt, position, water, humidity, Hall detector (magnetic), proximity (motion), and trespass detector to name a few. Schematics for these are included at the article link, along with the parts lists for each circuit.

You Have Mail

Quite often if you have a farm or even if you live in an urban area without home delivery, the mailbox is a fair distance from the house. Where I live, the mail is delivered to a drive-by mailbox only 50 feet from the house. However, there is a fair amount of theft that takes place with Social Security checks in our area. This makes it very important to get the mail as soon as it's delivered. Linx offers transmitters and receivers that they list as good up to 3,000 feet — a little over a half mile — which I utilize in this project.

The circuit presented here works both in collective

mailboxes and with standard lunch box style receptacles, and is adaptable. With a collective mail box, the device is mounted on the door with Velcro™ or double-sided tape. When the mailman opens the door, it detects light and activates the unit using a photodetector. It is important to mount the unit on the door so that it exposes the antenna and prevents a Faraday effect if placed in the back.

The unit transmits only one second and then goes to sleep. This trips the receiver which latches and sounds an alarm, then changes an LED from green to red. The alarm is reset by pressing the reset button on the receiver. You can use the same photodetector or a tilt switch in the lunchbox style mail bin.

No special tools are needed other than a soldering iron and a drill. The board files and schematics are generated by ExpressPCB software, which is available for free at www.expresspcb.com. Pre-programmed code matched microprocessors are available, along with boards from the *Nuts & Volts* webstore. Go to the article link for part numbers, sources, extra schematics, "Hints and Tips," PCBExpress board files, etc. **Figure 1** shows what's needed for our You Have Mail app.

The Transmitter and Receiver

The transmitter uses a Microchip PIC16F690 for processing several types of signals. A Linx TXM-418 transmitter is used which has a frequency of 418 MHz. A rotary BCD switch allows for 16 programs to be run; refer to **Figure 2**. It is important to remember when building a particular application that other parts can be omitted, e.g., if only the photodiode is used, you don't need the op-amp or its resistors or R5. However, more than one application can be run on the board since the additional parts will not interfere with each other. Keep in mind, the more applications, the more power the circuit will draw. The one exception is that only one light type sensor can be used at one time.

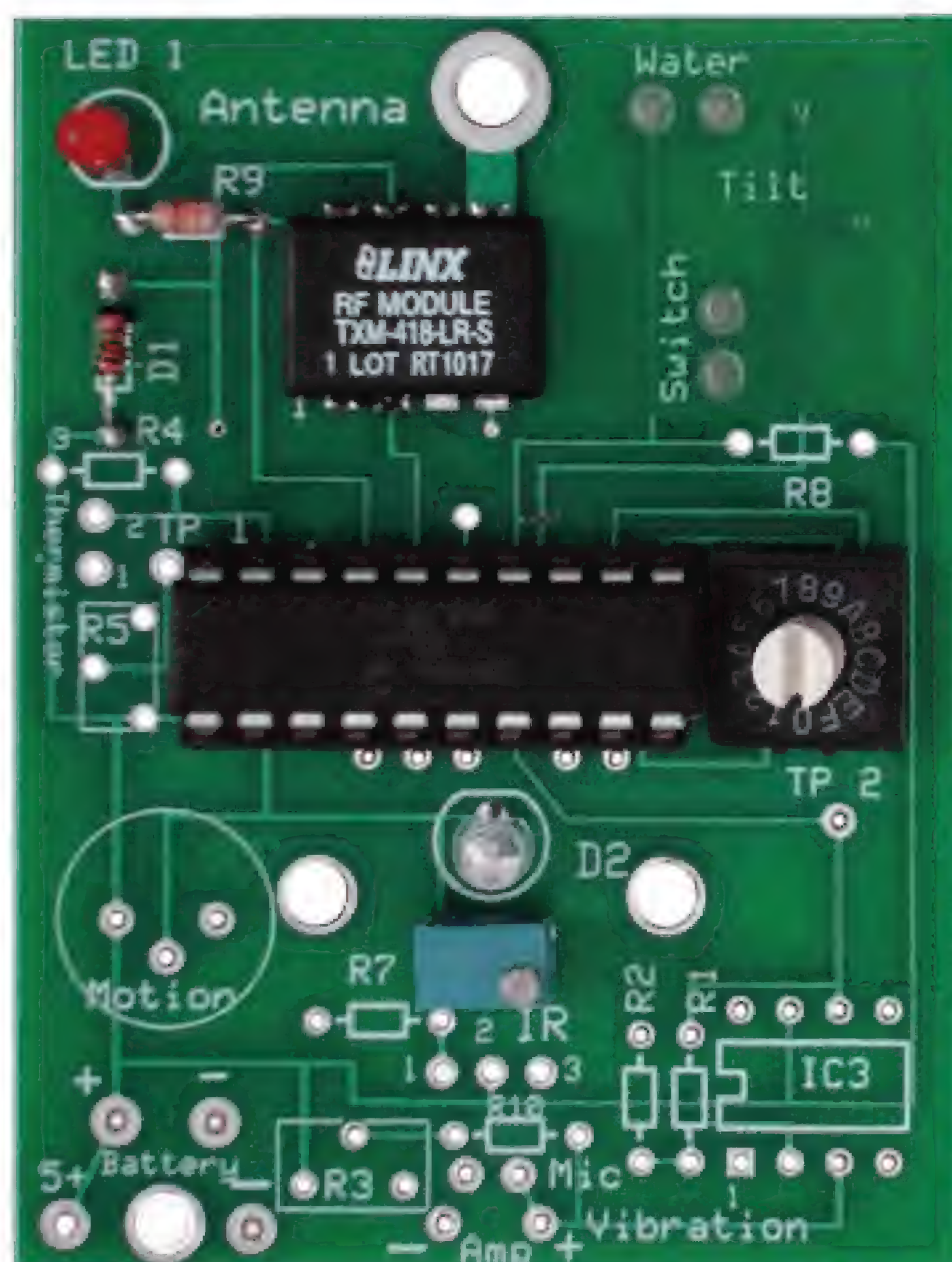
The transmitter microcontroller goes to sleep and is only awakened when a pin change takes place. It turns on the transmitter and sends a 256 code that activates the receiver. The method for modulating the frequency is bit banging. Any switch — whether normally closed or normally open — that changes will cause a transmission. The temperature and amplifier circuits are on the comparator microprocessor inputs. If they are above or below the voltage set by the reference voltage (R5), it will trip the micro. The above or below settings are determined by the rotary switch. I designed the transmitter software not to alarm when resetting its condition.

D1 provides a .8 volt diode drop to reduce the transmitter voltage to 3.3 volts.

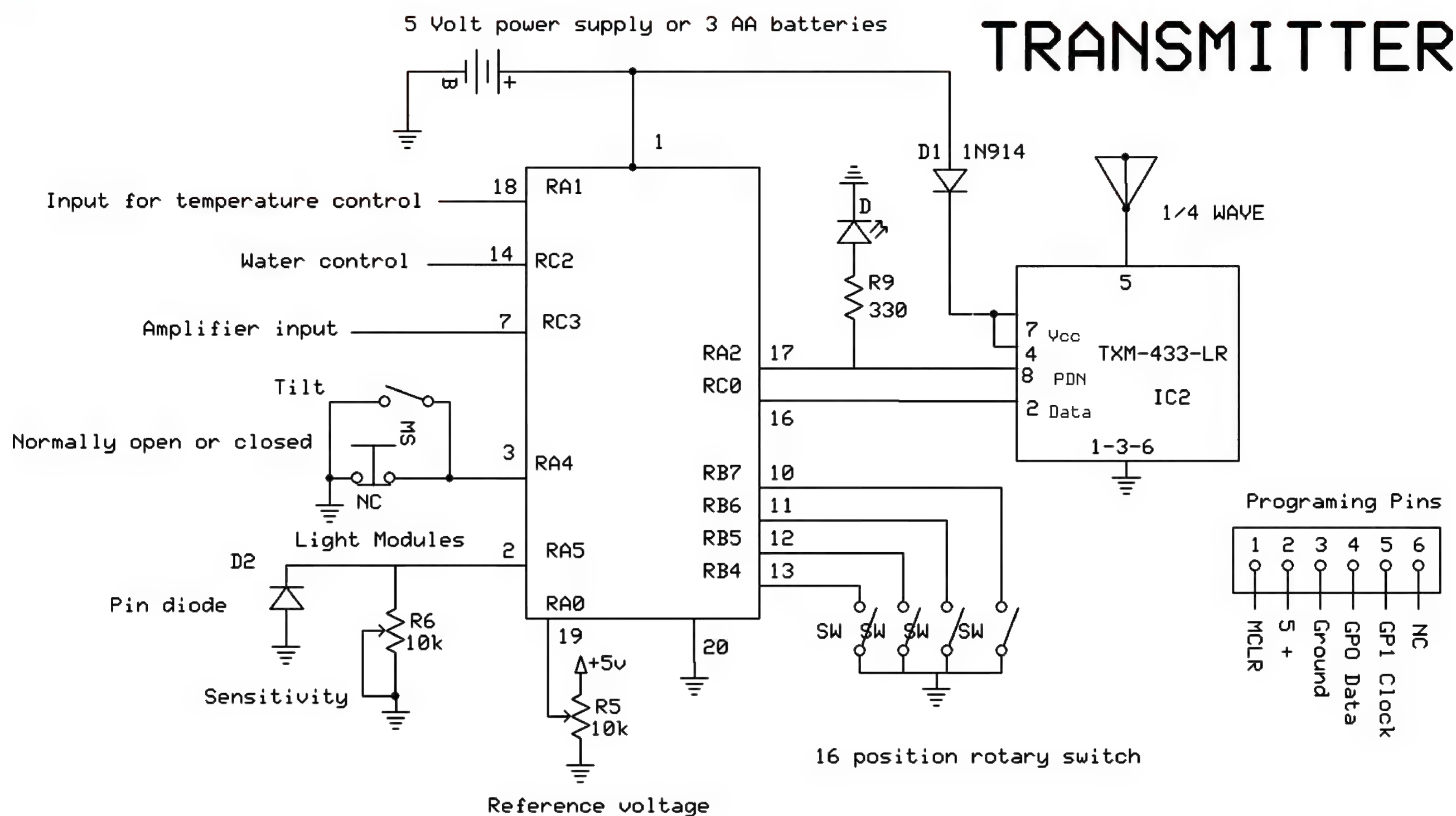
The receiver consists of a Microchip 12F508 and a Linx receiver RXM-418 chip; refer to **Figure 3**. The microprocessor decodes the transmitter codes and will alarm if it receives a signal matching its key code. It changes the color of the LED from green to red if a signal



■ **FIGURE 1.**



■ **FIGURE 2.**



has been sent, and the alarm will sound. The default is the “Latch” mode. The alarm will sound for about 10 seconds, then turn off. However, the alarm LED will stay red until the reset is pressed.

By holding down the reset switch for three seconds,

the “enunciator” yellow LED will turn on. The alarm will give a short beep and the alarm LED will change colors momentarily. For those who perform their own programming, again check out “Hints and Tips.”

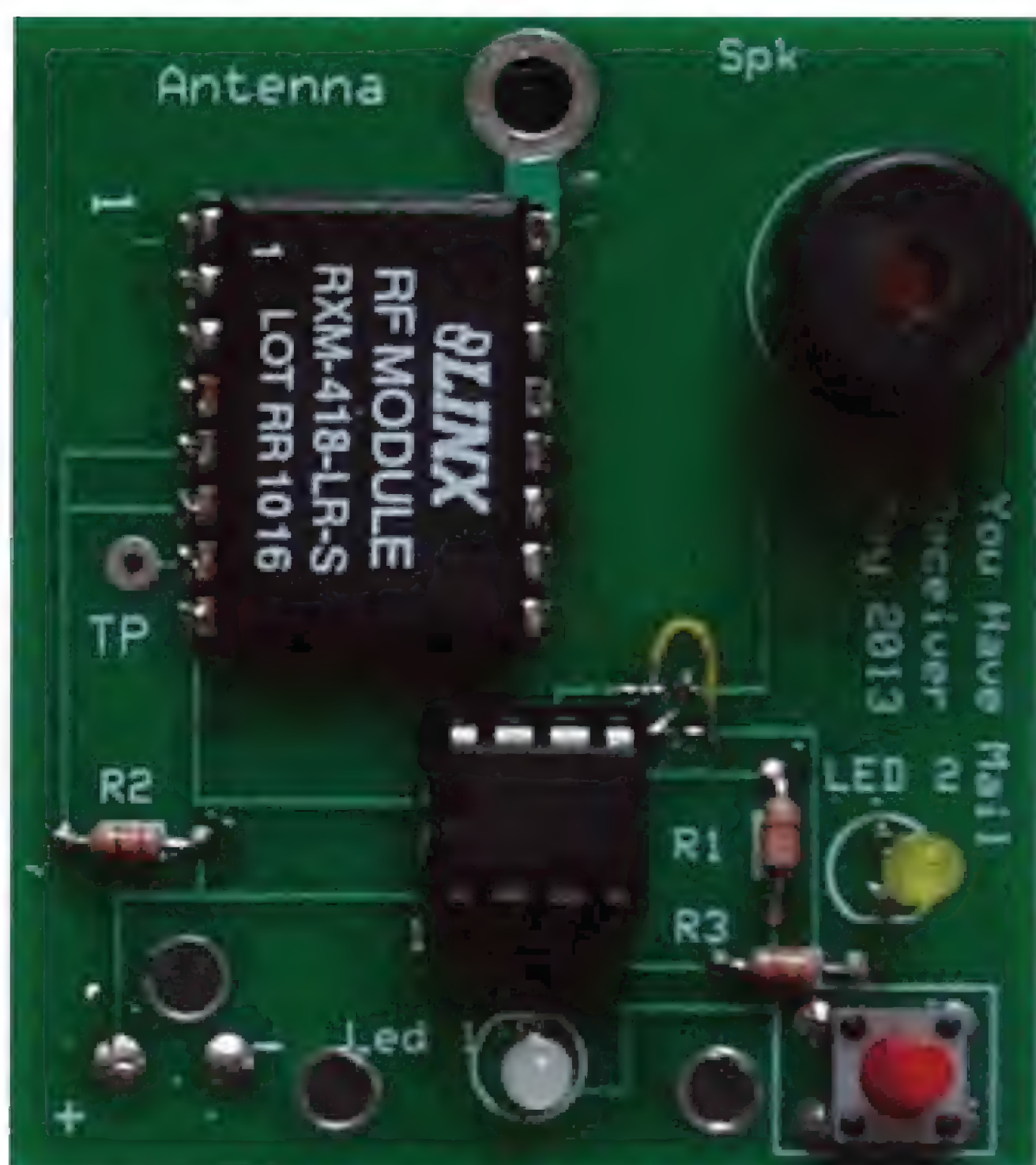
Building the Transmitter and Receiver

For both the transmitter and receiver, I have added

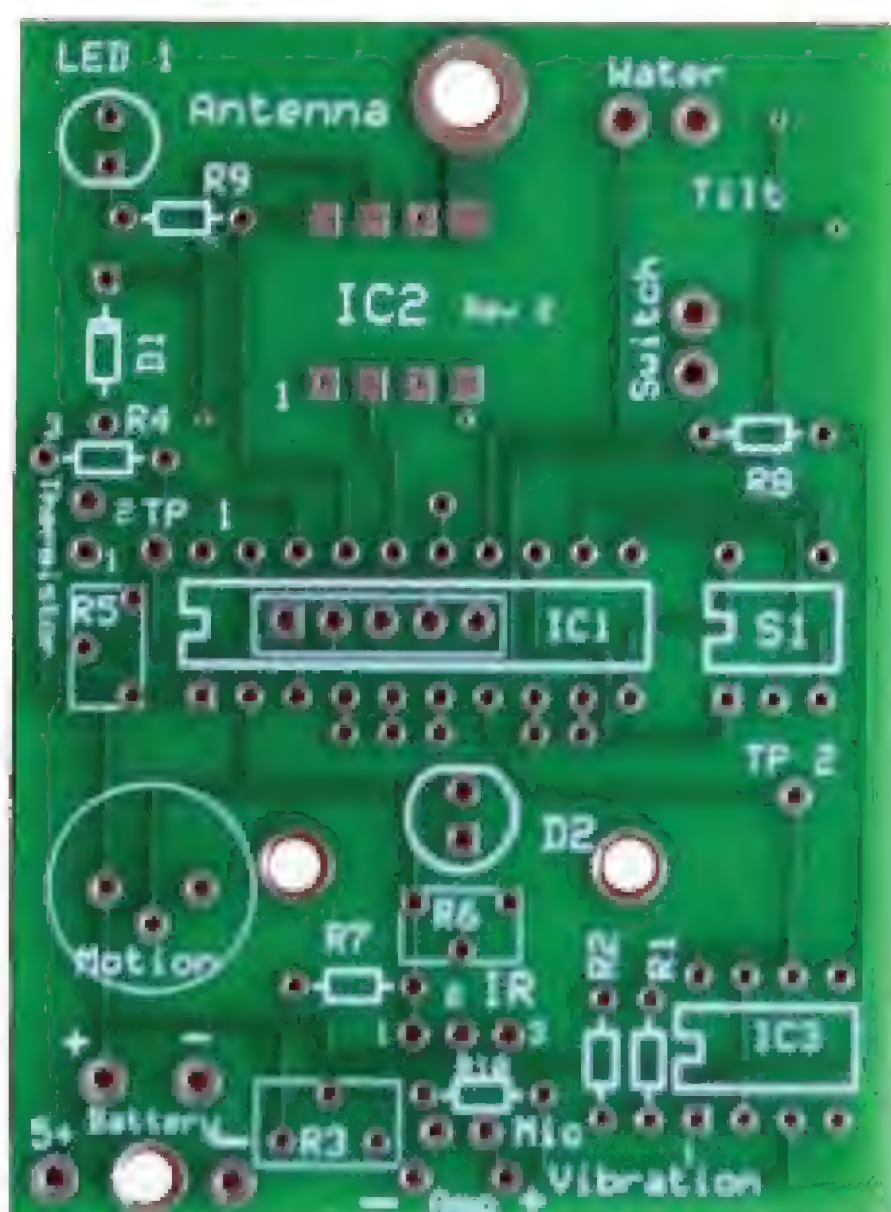
programming pins for the receiver’s PIC12F508 and the transmitter’s PIC16F690. This allows easy programming changes.

Refer to **Figure 4**. Determine what sensor or sensors you are going to be using. Solder IC2, first noting pin 1. Solder the socket for IC1 and then IC3 (if used), noting that the square pad is pin 1.

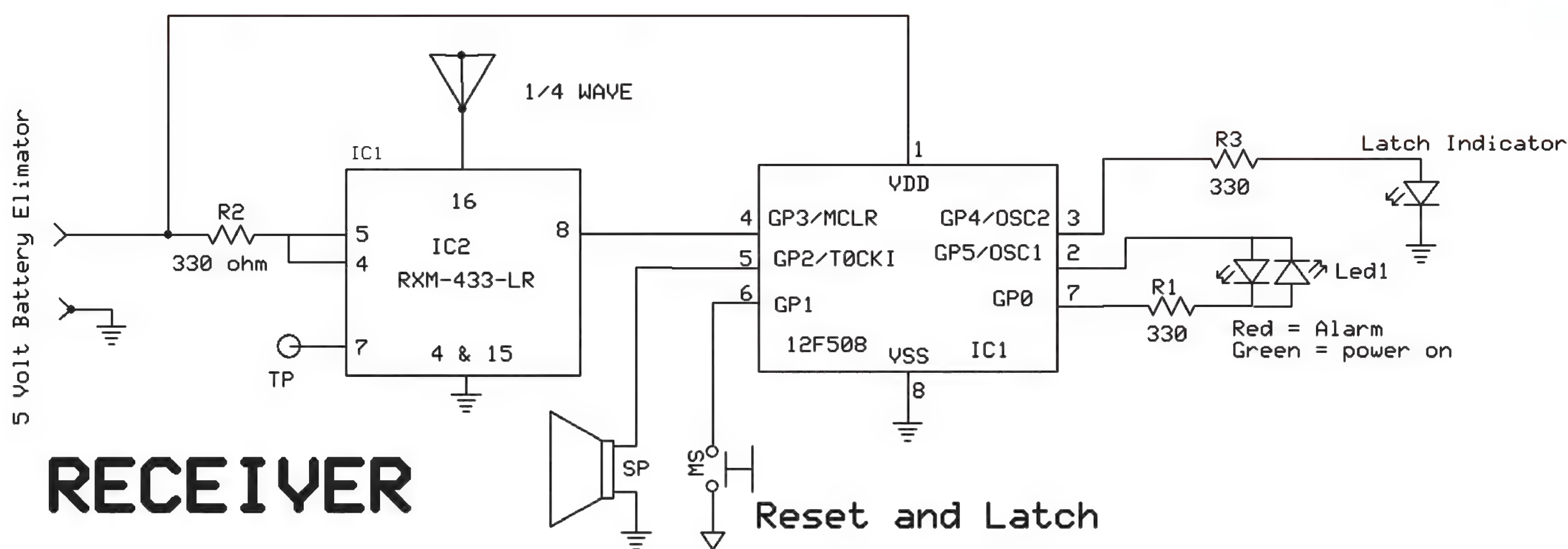
For the light mode, solder D2. Its long pin goes to the square pad and R6



■ **FIGURE 3.**



■ **FIGURE 4.**



and D1 (note polarity), and the rotary switch (note pin 1). Solder the LED and R9. Run the wires from the battery holder through the strain relief hole and solder red to + and black to -. If using different sensors, solder them in at this time. For the project shown here, we're also using the tilt switch. Locate the file called "Templates" at the article link. Download this file and cut out the templates. Glue the transmitter template to the top of the box using a glue stick. Drill the two mounting holes and the appropriate holes for the other sensors. Drill the antenna hole.

Put two 6-32 3/8" screws in the mounting holes from the front of the box. Add two 3/8" spacers. Attach the screws to the board. You will not need a nut because the board will self-thread. Place the antenna through its hole with the flat facing the back of the box. It is mounted to the top of the board; see **Figure 5**. Secure it to the board with its mounting screw.

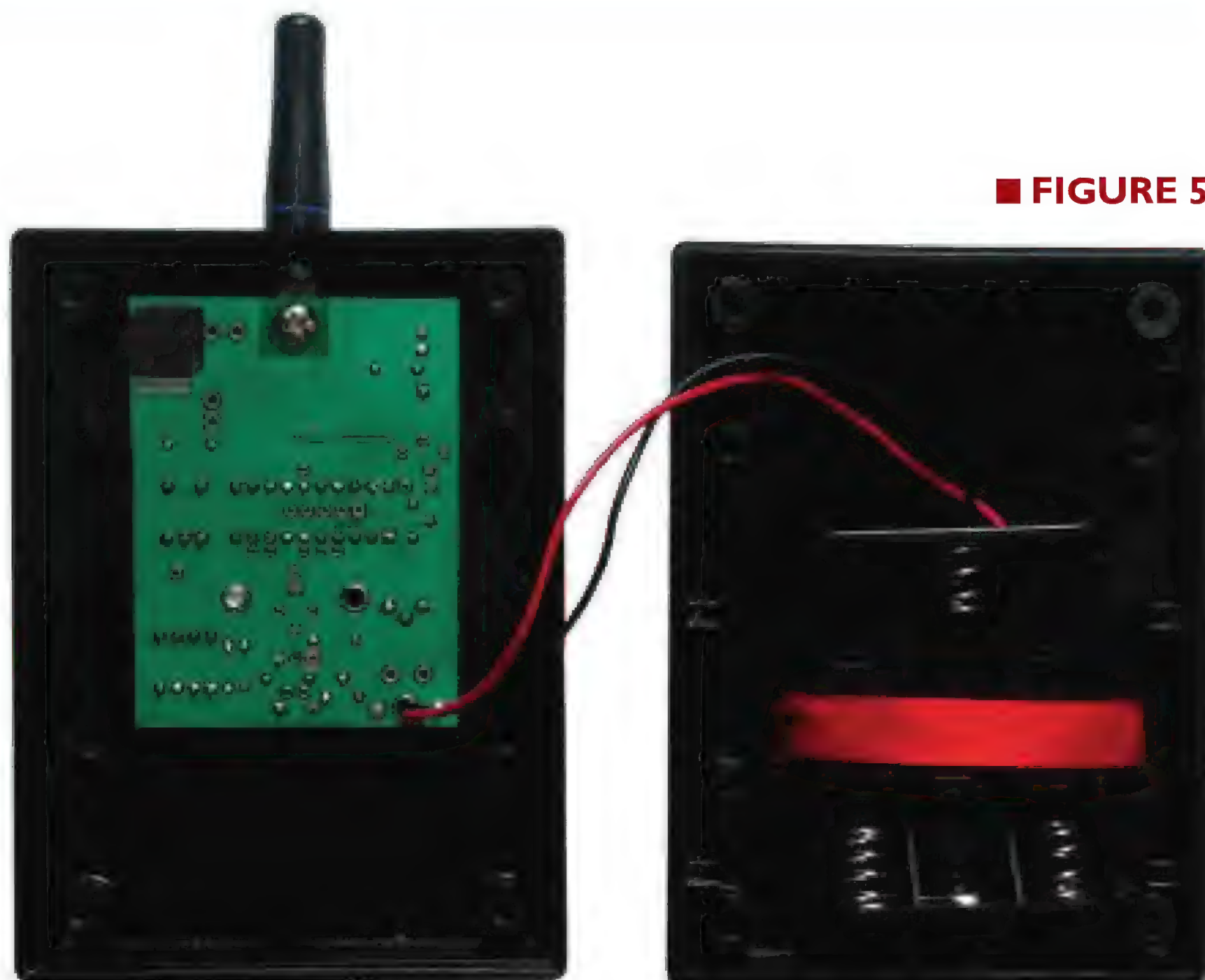
Light Detection for You Have Mail

Place the photodetector (D2), noting its polarity in the pads labeled D2. The anode (long pin) goes to the square pad. When light strikes the photodiode, it pulls the microprocessor pin to ground. Photodiodes have a very high resistance when no light strikes them. When photons hit them, the resistance goes low. Add R6 which adjusts the light

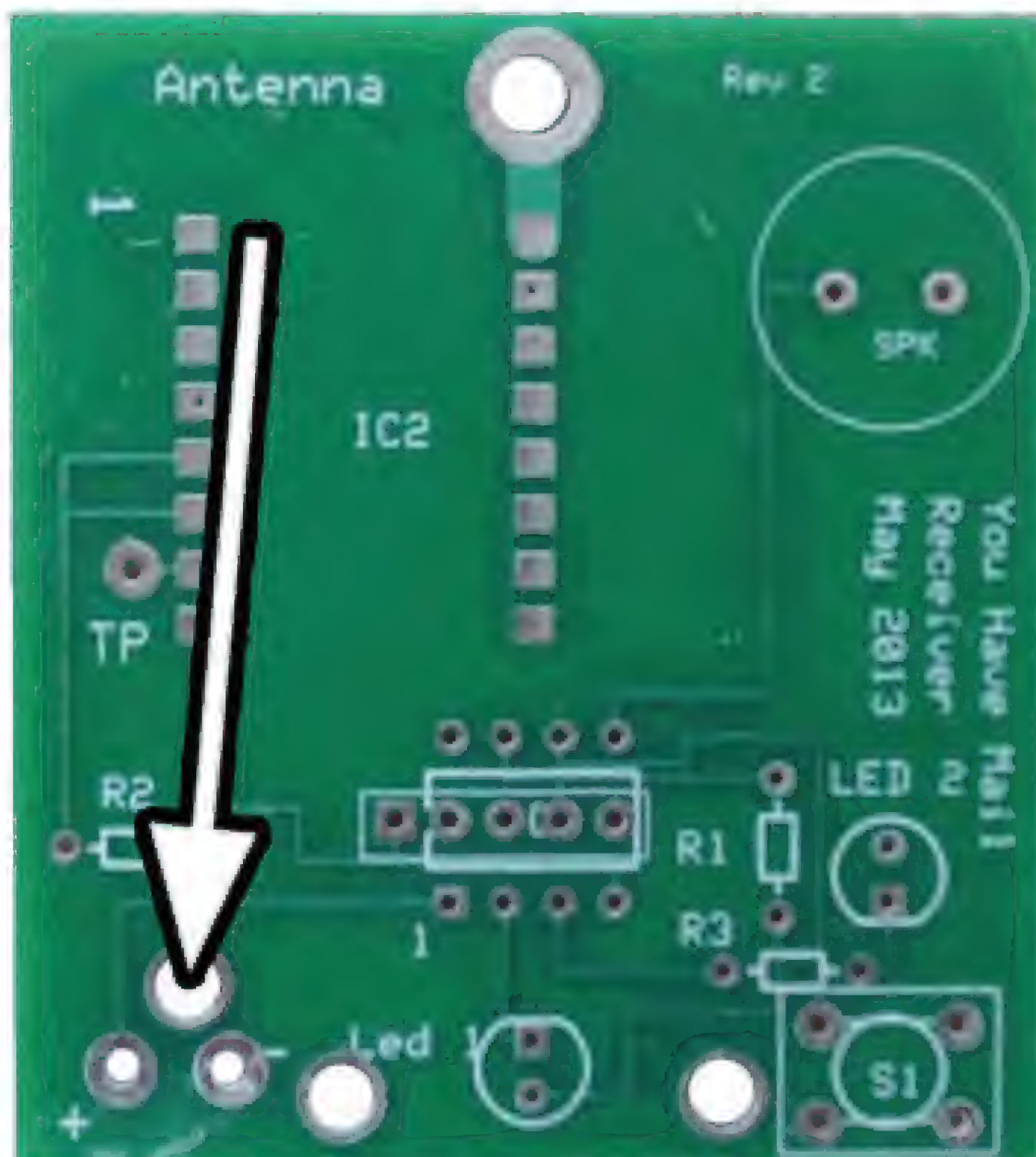
FUNCTIONS OF THE BCD SWITCH

0	Photo normally dark	Light mode
1	Photo normally light	Light mode
2	Proximity detects motion	Light mode
3	Proximity detect no motion	Light mode
4	Switch normally closed	Tilt or switch
5	Switch normally open	Tilt or switch
6	Switch change either NC or NO	Tilt or switch
7	Water sensor, no water	Water
8	Water sensor, water always present	Water
9	Amplifier alarms if above	Humidity, gasss, pressure, force
A	Amplifier alarms if below	Humidity, gases, pressure, force
B	Temperature alarms if above	Temperature
C	Temperature alarms if below	Temperature
D	Photo IR interrupted	Light mode
E	Photo IR detected	Light mode
F	Voltage change	Sound, vibration

■ **FIGURE 5.**



■ **FIGURE 6.**



sensitivity. If the unit is normally dark (in the mail box), set the rotary switch to the zero position. It will alarm when the detector senses light. If you want to detect when a room light goes out or the sun goes down, set the rotary switch to position 1 and it will alarm.

The Receiver

Solder IC2, noting pin 1. Solder the IC1 socket, switch, R1, R2, and R3. Add IC1. Solder the speaker.

Clip off the plug to the battery eliminator, thread through the strain relief hole, and solder the positive line to the + and the negative lead to the -. Refer to **Figure 6**.

Put two 6-32 3/8" screws in the mounting holes from the front of the box. Add two 3/8" spacers. Put the LEDs with the long lead to the square pads and leave them loose. Attach the screws to the board. Mount the antenna as described above. Push the LEDs into their holes. Now, solder the LED leads to the board. This will give the correct distance for mounting the LEDs.

Let's See If It Works

Add three AAA batteries to the transmitter and plug in the receiver. The receiver LED may be red or green.

1. Put the transmitter in a dark box.
2. Push the reset button on the receiver. The LED should be green and the alarm should be silent.
3. Take the transmitter out of the box and into the light. The receiver should beep and the LED should turn red. The beeper should stay on for about five seconds. Press the reset button to change the LED to green. Once the transmitter is placed in the dark, it will reset.

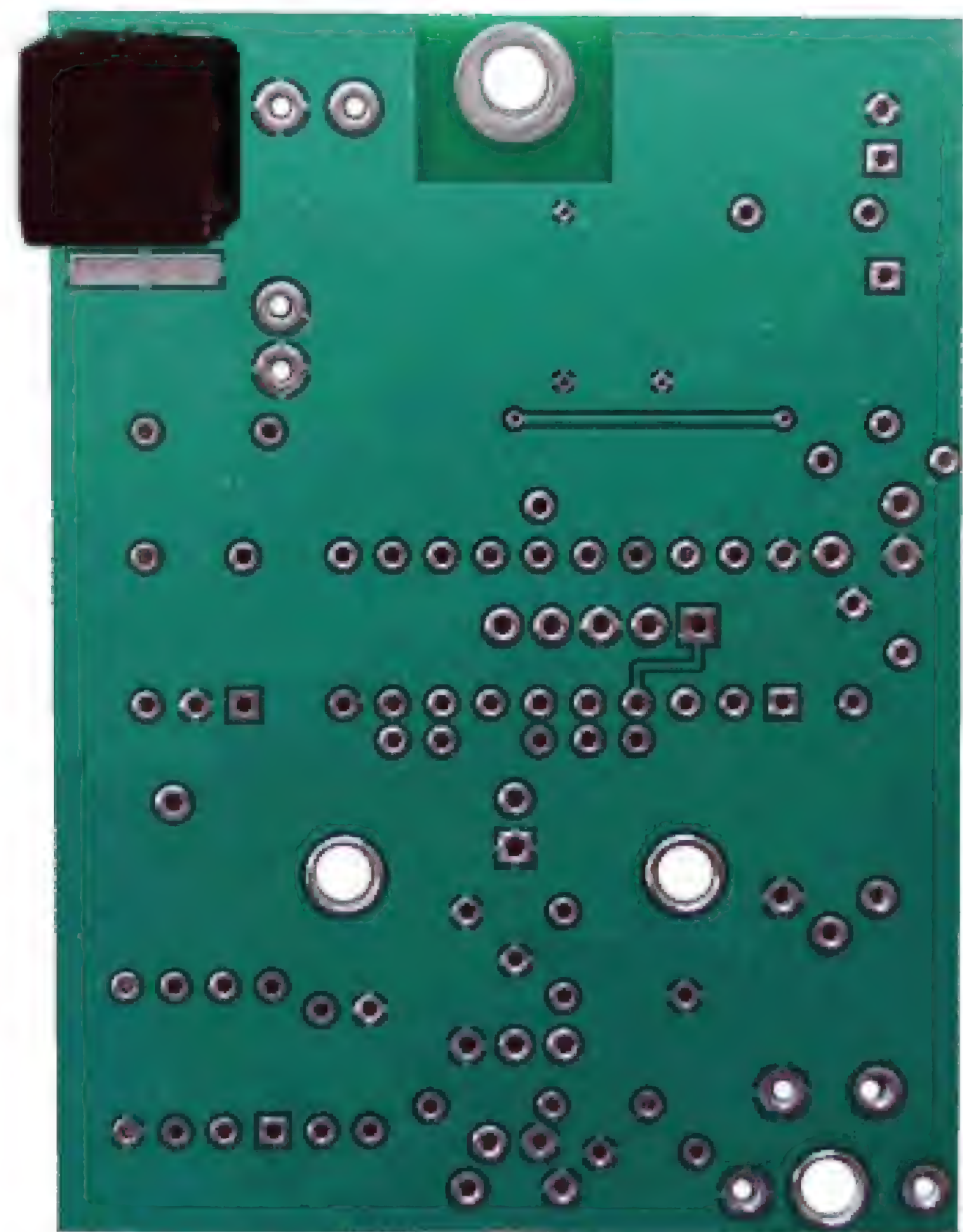
If you are using the trespass or the proximity detector, you may want to use the enunciator mode. Hold down the reset button until the enunciator LED lights. To reset to the alarm mode, hold down the reset button until the enunciator LED goes off.

The unit can be built with several applications on the board that can be changed with the rotary switch. However, the light, proximity, and trespass use the same circuit so only one can be used at a time.

Tilt and Switches

The tilt function was designed using an OncQue RBS070500T tilt sensor. It can be mounted horizontally or vertically using the pads labeled "tilt." Check the specification sheet of the tilt switch for how different sensors perform. Mount the tilt on the bottom of the board as shown in **Figure 7**.

■ **FIGURE 7.**



There are two types of switches that can be used: normally open and normally closed. The unit can be programmed for closing or opening any switch to activate the transmitter. The software is written for both types of switches as there may be applications for both types. Solder the switch to the pads labeled “switch.”

Set the rotary switch to four for detecting the opening of the switch. Set the rotary switch to five for detecting the closing of the switch. Set the rotary switch to six to detect if the switch has changed from either open or closed (toggle mode).

The tilt and switch modes use RA4 which is programmed to detect a pin change using the interrupt mode. Any change will cause an interrupt. The software determines if it was normally closed or normally open.

OTHER APPLICATIONS

Water

A water sensor can be made using two stainless screws and a PVC plug. Again, refer to “Hints and Tips” for more information. You can make a flat sensor by taking a 1” square of circuit board and removing a 1/16” strip of copper down the center and soldering the wires to two rectangles. These are connected to the one and two inputs. Use a 100,000K resistor in R4. The conductivity of water is well below 100,000 ohms. R8 and the sensor act as a voltage divider. The comparator checks voltage across the sensor and compares its voltage to the internal reference voltage. If wet, the voltage will be below the reference voltage and trip the transmitter.

The water control employs its own circuit (RC2) using a low internal reference voltage. Place the rotary switch to six for setting off the alarm when the sensor detects water. For measuring water in a vessel, place the rotary switch to seven to be notified when the water level is below the sensor. For detection of lack of water, set the rotary switch to eight.

Amplifier and Other Types of Inputs

For any sensor that generates in the microvolt, millivolt, or volt range, you can use the amp input pads. The amplification is calculated by the following formula:

$$Amplification = 1 + (\frac{R1}{R2})$$

This amplifier can also be used for flow sensors or pressure sensors which generate small voltages. For sensors that produce voltages up to 4.5 volts, just eliminate R2 and short R1 to avoid any amplification. Do not exceed 4.5 volts on the input. Use an eight-pin socket for IC3 just in case you blow the MCP601. You will need R5 for setting the voltage reference. The reference voltage can be checked at the pad labeled TP1. The amplifier output voltage can be checked at TP2. The amplifier uses the RC3 input of the comparator of the micro. The amplifier output (TP2) is compared to the voltage generated by R8 (TP1). Any change causes an interrupt. The software determines if the voltage is above or below the reference. Set the rotary to eight for voltages to trip above the reference voltage (TP1, R5). Set the rotary to nine for voltages to trip below the reference voltage (TP1, R5).

Vibration

The vibration sensor is a MEAS piezo film sensor with a weight. Mount it hanging from the board by bending its connectors, and solder it to the two pads labeled “amp.” Add a one megohm resistor to R10. Solder in R3 and set TP2 to about 2.5 volts. Adjust R5 for the tripping voltage

ITEM	DESCRIPTION	PART #	SOURCE
TRANSMITTER:			
Antenna	418 MHz 1/4	712-ANT-418-PW-LP	Mouser
Battery box	Three AA	12BH331-R-GR	Mouser
Box	Plastic	SR131B-ND	Digi-Key
D1	1N914	1N914BCT-ND	Digi-Key
IC1	PIC16F690	PIC16F690-I/P-ND	Digi-Key
IC2	TXM-418 LR	712-TXM-418-LR	Mouser
LED 1	Red 3 mm	754-1606-ND	Digi-Key
R9	330 ohm 1/6	330EBK-ND	Digi-Key
Socket	20-pin	AE9998-ND	Digi-Key
S1	Rotary switch	106-H2A16-EV	Mouser
RECEIVER:			
Antenna	418 MHz 1/4	712-ANT-418-PW-LP	Mouser
Battery eliminator	Five volt	T977-P6P-ND	Digi-Key
Box	Wall mount	SR031-WIB-ND	Digi-Key
IC1	PIC12F508	PIC12F508-I/P-ND	Digi-Key
IC2	Linx receiver	RXM-418-LR	Mouser
LED 1	Bi-colored	160-1058-ND	Digi-Key
LED 2	Yellow 3 mm	754-1602-ND	Digi-Key
R1-R2-R3*	330 ohm 1/6	330EBK-ND	Digi-Key
S1	19 mm momentary	EG1866-ND	Digi-Key
Socket	Eight-pin	AE9986-ND	Digi-Key
Speaker	4 kHz	445-2525-1-ND	Digi-Key
LIGHT DETECTION:			
D2	Pin diode	EL-PT204-6C	Mouser
R6	10K trimmer	490-2970-ND	Digi-Key
TILT DETECTION:			
S2 Tilt	Tilt switch	108-07050T	Digi-Key
*All parts one each, except R2-R2-R3 — three each are needed.			

PARTS LIST

using TP1 to just above or below TP2. For R1, use a 100K ohm resistor and a 10K ohm resistor for R2. Set the rotary switch to F for sounding the alarm when a vibration occurs.

The vibration sensor feeds into the non-inverting input of the op-amp which was described previously. It is amplified about 11 times and generates a voltage when moved or jarred. R10 adds the voltage generated from R5 to the amplifier. The piezo film generates a small amount of voltage when bent. R8 is set just above or below the voltage of the output of the amplifier. When the sensor vibrates, it will cause an interrupt.

Sound

For sound applications, use a PUI POM-5238 microphone and solder it in noting the polarity. Use a 100K trimmer resistor for R3; R3 sets the sensitivity. Short R10 with a jumper. The amplification can be adjusted by changing R1 and R2. When sound pressure changes, the microphone changes the voltage out. Measure the voltage of TP2 and adjust R3 to about 2.5 volts. Adjust R5 for the tripping voltage using TP1 to just above or below TP2. The sound mode functions the same as the vibration mode.

Set the rotary switch to F for when a sound is detected.

Proximity/Motion Detector

The motion detector was designed for use with any of the new Panasonic motion detectors. The cost of the sensors is about \$9.50. They are very small and can measure movement from distances of 10 meters. Honeywell makes several types of digital motion detectors. Using the template, drill out the hole called for. Solder in the transistor socket into the pads labeled "motion." Add a one megohm resistor, R7. Mount the board to the chassis using only one screw. Put the sensor through its hole and plug it into the socket. Set the rotary switch to two to activate when motion is detected. Set the rotary switch to three to activate when no motion is detected.

The Panasonic motion sensors are pyroelectric sensors that detect variations of infrared. They send the pull-up resistor of RA5 to ground when detecting heat. R7 biases their output. They take about 30 seconds to stabilize.

Temperature

Use a 10K thermistor across pad 1 and pad 2. This can be mounted remotely. Add a 10K ohm resistor in R4. The R5 potentiometer is used for the tripping point by varying its output voltage. There is a listing of voltages of the thermistor vs. temperatures in Centigrade available at the article link. The temperature can be changed when it is either too high or too low by switching the rotary switch. You can measure the tripping voltage by using TP1.

Set the rotary switch to B for sounding the alarm when above a set temperature. Set the rotary switch to C for sounding the alarm when below a set temperature.

With this circuit, you can substitute any resistor for the thermistor and adjust the pot for the tripping point. Many sensors use a change of resistance. Position sensors are one of these. Just connect the sensor across pads 1 and 2 with an equal resistance on R4.

Some position sensors have three terminals: positive, negative, and wiper. Eliminate R4 and place the sensor with the wiper on pad 2 and the other two terminals on 1 and 3.

I often make up temperature probes by using two strands of #30 wire wrap up to 20 feet in length. The resistance of the sensor is 10K ohms at 20 degrees C, so the wire resistance is negligible. Twist the wire together and wire wrap to the thermistor, and solder. Take a red plastic twist connector and remove its metal core. Push the thermistor into the cap and use a hot glue gun to seal the thermistor. This makes it waterproof. The temperature software has built-in hysteresis to prevent chatter. It uses the voltage comparator of RA1 and RA0 with an interrupt.

Trespass Detector

This is one of my favorites due to the distance attainable. The trespass detector uses an infrared detector with a high power infrared-pulsed emitter. It is basically an invisible light beam that if broken, sounds an alarm.

Bend the leads of a Vishay TSOP2436 IR receiver and place them into the three pads labeled "IR." The lens should be in the circle where the photodiode would go. R6 will not be present. The specifications on these two devices indicate they may have a range of up to 45 meters (135 feet).

You will need a 110 volt AC to five volt DC power supply on the IR source as it is continuously on and draws over 200 mA. Set the rotary switch to D to sound the alarm when the IR beam is broken. Set the rotary switch to E to activate when the IR beam light is sensed. The IR transmitter and receiver have to be the same frequency to work properly. The IR transmitter is pulsed at a certain frequency to avoid false signaling from other light sources.

The IR transmitter uses four 10 degree TSAL6100 IR emitters pulsed at 400 mA. The first one I built had a dropping resistor which got hot. Then, it dawned on me ... why not use four IR emitters for a broader beam and let them perform the voltage drop? The IR emitters are pulsed at a 36 kHz frequency for .3 milliseconds and then turned off for 10 milliseconds. The receiver is constantly on and functions as a missing pulse indicator. If the beam is not detected for more than 20 milliseconds, it alarms. Since the unit is constantly measuring, I recommend a five volt power supply to prevent the battery from running down. Keep in mind that the receiver can be a latched alarm or will just beep when the beam is broken.

The receiver pulls RA5 to ground when an IR source at

the right frequency strikes the receiver.

Building the Trespass Transmitter

Solder the preprogrammed PIC 12F508 (available from the *Nuts & Volts* webstore, along with the board). Note pin 1 is the square pad. Solder the R1 (10K) resistor and Q1. Solder the four IR emitters, placing the long lead into the square pads.

Locate the IR transmitter template; cut and glue it onto the box lid. Drill the proper holes for the LEDs. Drill a hole for the cord. Cut off the plug of the battery eliminator and pass it through the side hole in the box. Determine the positive wire using a voltmeter and solder it to its pad on the board. Solder the other wire to the negative pad. Mount the board to the lid using two 3/8" spacers and two 6-32 1/2" screws. Note: The LEDs will not show a visible light when plugged in.

Humidity

The TDK CHS-MSS is a five volt sensor that outputs 0-1 volts for 0% to 100% humidity. Its output can be placed on the "amp" inputs. Pull its five volt supply from the pad above the battery terminal. For maximum sensitivity, use a five amplification with the op-amp; R1 = 100K; R2 = 24K. There is a drawing at the article link for measuring the humidity of the Earth.

Set the rotary to nine for voltages to trip if the humidity is above a set point. Set the rotary switch to A for voltages to trip if the humidity is below the set point.

Calculate the voltage of the sensor and multiply it by the amplification. Set the R5 voltage to this calculated voltage using the test point. Use three strands of #30 wire wrap to use the humidity sensor remotely. Five volts+ can be obtained from the extra pad next to the battery input. Make sure the ground lead goes to the - pad of the amp input and the sensor out lead goes to the + pad.

Gasses

Parallax makes a number of gas detectors and a board for their processing. I have put extra pads for a power supply for driving another board. Gas detectors pull a fair

Future items you might want to add for monitoring and alarming include: gasses (O₂, CO₂, gasoline, methane, propane, alcohol, etc.), or a radiation monitor. These will need an additional board to function. I have added extra pads for those who wish to program and add to this project. I only used the comparator portion of the chip; however, the chip has A/D converters. The possibilities are limitless. Keep in mind that the range of the radio transmission is up to 1/2 mile. Monitoring of many sensors is now possible *without* wires! (Please contact the author for details on these add-ons.)

amount of power as they use a heating element. Make sure you use a battery eliminator because they will drain a battery in a matter of hours. Use the amplifier pads for input, and adjust R1 and R2 for amplification.

Set the rotary to eight for voltages to trip above the reference voltage (TP1, R5). Set the rotary to nine for voltages to trip below the reference voltage (TP1, R5).

Pressure and Force Sensors

Jameco carries a number of reasonable pressure sensors. These require a five volt power supply and can be interfaced to the "amp" inputs. Most pressure and force sensors use a strain gauge in a Wheatstone bridge configuration, and their voltage changes with pressure. Use the amplifier pads for input, and adjust R1 and R2 for amplification. Set the rotary to eight for voltages to trip above the reference voltage (TP1, R5). Set the rotary to nine for voltages to trip below the reference voltage (TP1, R5).

This versatile circuit provides tons of applications. I'm sure you'll be able to implement it in many of your own projects. **NV**

DIY NEVER HAD A BETTER FRIEND...

Take your DIY project to the next level with PanaVise work holding tools. Perfect for hobbies, soldering, electronics, crafting & repair, PanaVise tools have hundreds of uses only limited by your imagination.

PanaVise, "better than a third hand!"

PANAVISE
Innovative Holding Solutions

Now available at
RadioShack

Model 301
640-0178

Model 324
640-0177

www.radioshack.com/StoreLocator

BUILD THE LED MATRIX GAME PLAYER



By Matthew Bates
mattscottbates@gmail.com

Go to www.nutsvolts.com/index.php?/magazine/article/october2013_Bates for any additional files and/or downloads associated with this article. You can also post comments.

Those 8 x 8 matrix displays are good for more than just making scrolling signs. With a little imagination and a whole lot of extra time on your hands, you can actually use them to create simple yet amazing little action games. No matter how many or how few pixels you have at your disposal, the theme remains the same. It is always a matter of destroy or be destroyed, chase or be chased, or something along these lines. If the game is interesting enough, at some point you stop noticing that it is taking place on a field of only 64 LEDs. The only thing that matters is victory.

Matrix Displays

The most common, simple, and least expensive matrix display is an organization of 64 single or dual color LEDs arranged as eight rows by eight columns. The footprint standards for these displays are typically referred to as small, medium, and large, ranging in size from 3/4 to 1.5 inches square. These LED displays come in two distinct polarity types defined as Anode in Row or Cathode in Row, which denotes how the rows are connected to the diodes.

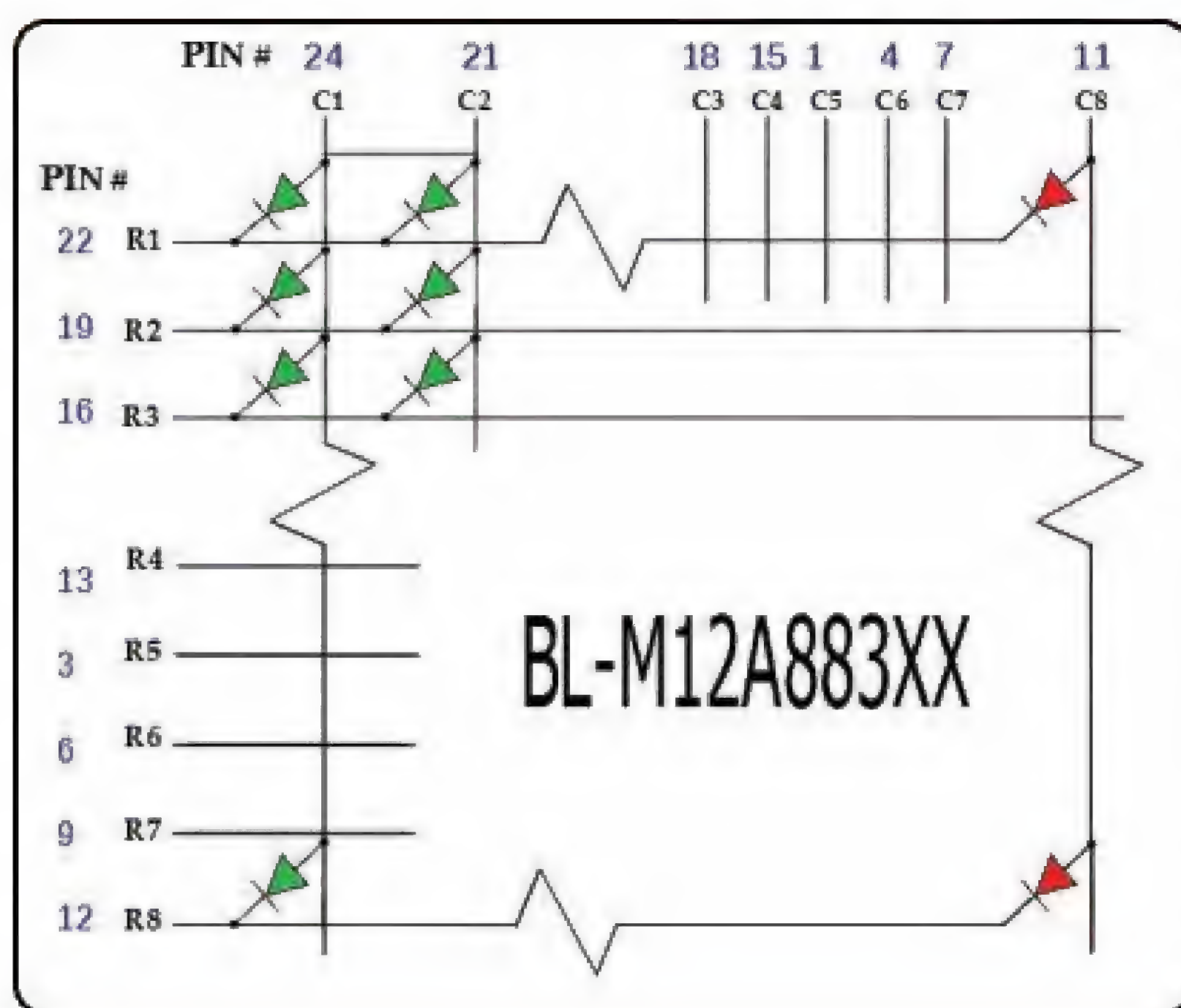
This polarity definition is important at a programming

level, by defining whether a low or high command is issued to light the individual LEDs. The illustration in **Figure 1** shows a common organization in which the rows are all connected to the common cathode side of the diodes.

The 24-pin dual color displays are all pretty standard as far as pin numbering. I find that the square pattern LED is a better choice for most game play. The square pattern display seems to be less common than the traditional round pattern. Adafruit is a good source for high quality dual display units but they can be had from various places. Futurlec sells a less expensive 16-pin single color display.

All that is needed to illuminate any individual (or combination of) LED is to bring a single or multiple set of rows high, and to strobe the associated column connection to turn the diode(s) on or off. Each LED will illuminate brightly at 3 mA when pulsed at around 50 msec. To discretely turn all of the 64 LEDs on or off will require at least 16 I/O lines from any common microcontroller, or eight port control for the rows and a shift register to control the columns.

In the event that all the LEDs must be turned on or appear on, the device driving the LEDs must be able to source 3 mA x 64 without using a buffering scheme such as discrete transistors or a transistor array package. The PICAXE 28x2 has a total of 20 bidirectional ports and has a maximum current rating of 200 mA. By designing a display driver using only a low voltage microcontroller, we can build a game that can run at least eight hours between charges with a single lithium battery rated at less than 200 milliamp hours.



■ FIGURE 1. Diode connections for a dual color matrix.

How the Design Works

Any game worth playing will require at least three buttons for control. Two of the three buttons should be dedicated for directional maneuvering. A third button is essential for actions such as firing control, and can be used for starting the game. The addition of sound capability is also a must for any game design.

Without using a shift register for row control, 16 I/O ports are used for the display. Three ports are used for pushbutton switch input and the remaining single port is used for audio, for a total of 20 I/O ports. Only the display and pushbuttons are located on one side of the board as illustrated by the schematic shown in **Figure 2**.

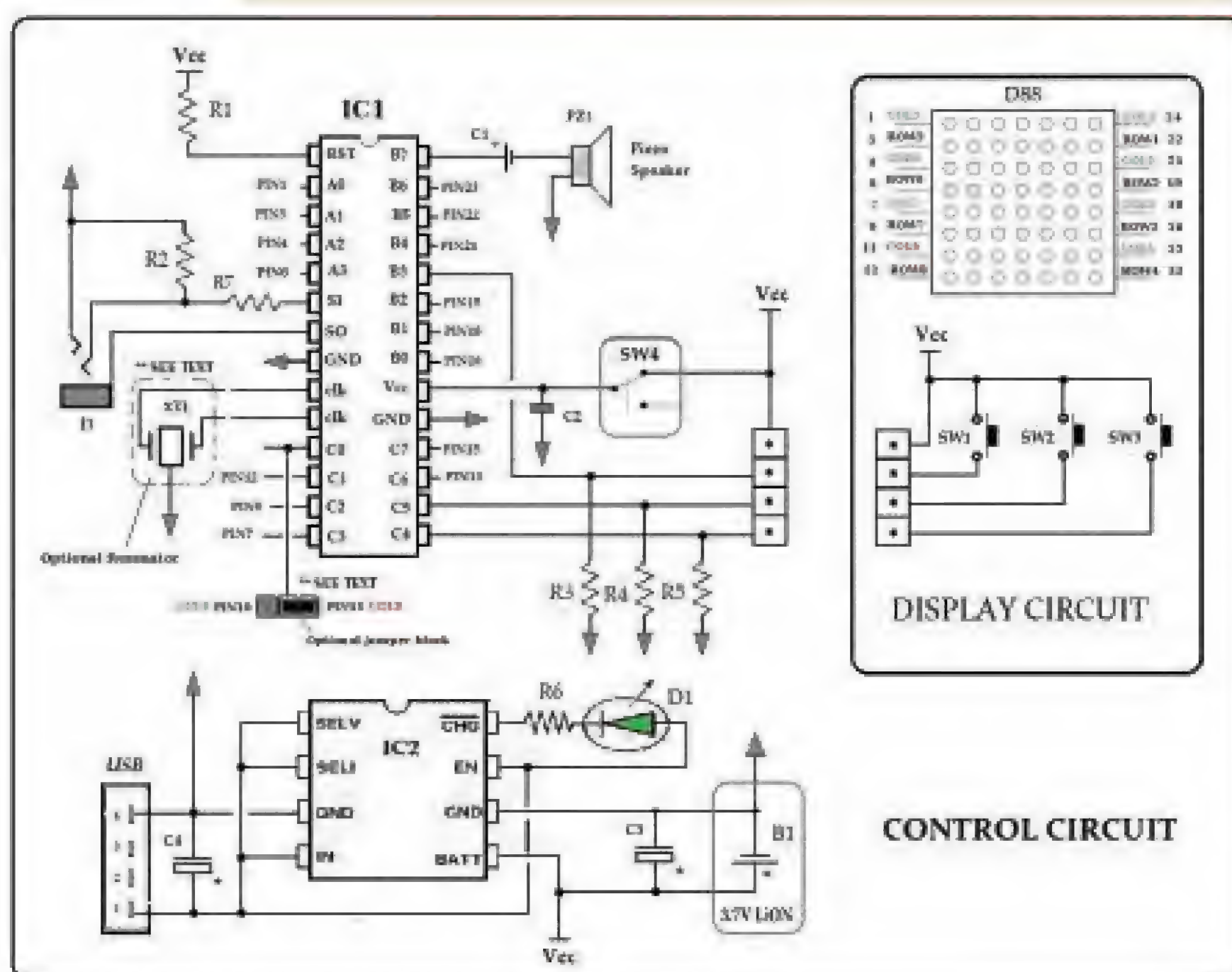
The three pushbutton switches connect to the upper side of the board. I purposely located the center pushbutton switch offset lower to the other two directional buttons. This is a subtle change that makes a difference given the function of the switch.

You may notice that the eighth column for the display is connected to the red columns of LEDs. This was done specifically for the game that is included with this article. Row 8 is the lower horizontal array and is used for the player

paddle construct. You may want to include a jumper to select the green or red color for this column in future games, as shown in the connection to port C.0.

The bottom side of the board holds the lithium battery, a lithium cell charging circuit, the USB-A female socket, and the PICAXE programming port. The lithium

■ FIGURE 2. Circuit schematic for the project.



battery used is a 3.7V, 140 mAh lithium polymer type which will survive at least eight hours of constant use. This battery is really small with a footprint of only 34 mm x 20 mm x 4 mm.

The **schematic** shows how the PICAXE connects to the other components of the system. As you can see, the display is connected to 16 bi-directional ports on the PICAXE. The three SPST tactile switches connect to three of the PICAXE ports, and are all configured with 10K pull-down resistors with the button tied to the supply voltage. Connecting the buttons in this fashion will present a logic high to the associated pins when they are depressed.

The piezo element speaker is connected to port B.7 and is interfaced using a 10 µF electrolytic to increase the sound volume. The two programming pins — serial IN and serial OUT — are connected to a stereo jack. This is the typical connection method for programming the PICAXE,

■ **FIGURE 3. Mounting the MAX1811 to the SOIC to DIP adapter board.**



converting USB to serial. The 28X2 PICAXE only requires an external crystal to run at frequencies higher than 16 MHz.

The board has been designed to add a resonator later if needed. A miniature right angle slide switch is used for on/off control by breaking the connection to the 3.7V battery supply for the circuit. The circuit is configured to begin charging the 3.7V lithium battery once power is supplied to the USB connector. Charging is accomplished by way of the MAX1811 single cell lithium battery charging manager. The MAX1811 can be configured to charge either 4.1V or 4.2V standards, and at selectable current settings.

The **schematic** connects the MAX1811 to deliver 500 mA for charging. The MAX1811 displays charge status by lighting an LED at the 2.4V mark and extinguishes once the battery is charged. Unfortunately, the MAX1811 is only manufactured as a surface-mount part and has to be soldered onto an SOIC to DIP adapter board. The illustration in **Figure 3** shows how this is accomplished.

I find that it helps to hold the chip in place with an alligator clip or by using double-sided tape before you begin soldering the part to the board. Just be sure to use an adequately heated soldering iron (375 deg) with a proper tip for the task.

Once adapted to DIP, the board can be soldered to the circuit board or in an eight-pin socket.

Designing the Circuit Board

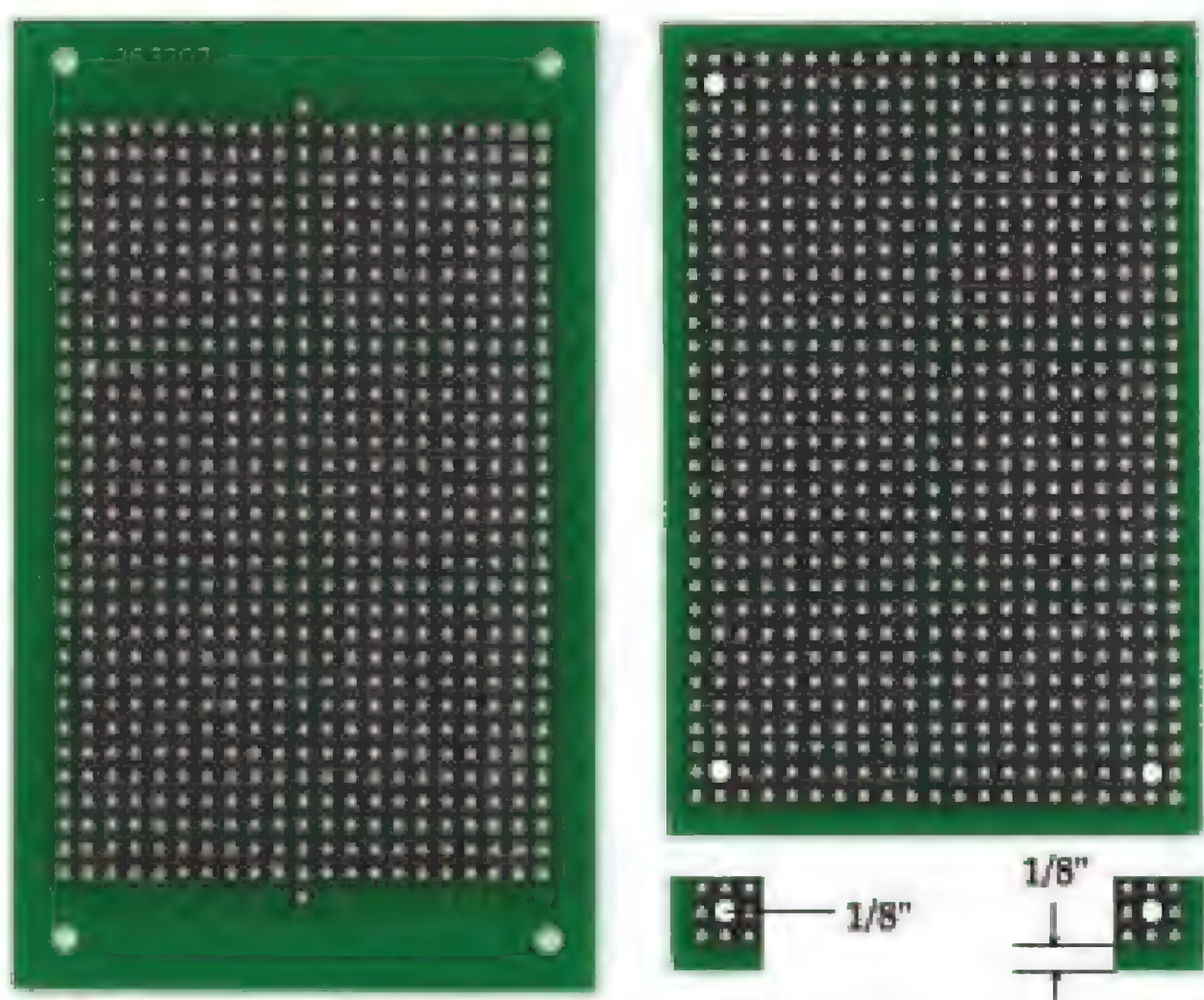
To construct the circuit board for this project, I used a double-sided plated through hole board I purchased from Jameco. I modified the board slightly with a Dremel tool by reducing the space to the sides for mounting the programming socket and miniature right angle slide switch.

Without trimming the sides, the audio socket and slide switch are too far from the edges. It is a good idea when designing a circuit like this to actually assemble the pieces first before soldering the parts. The illustration of **Figure 5** is an example of an assembly using the standoffs

ITEM	QTY	DESCRIPTION	SOURCE/PART #
IC1	1	PICAXE 28X2 controller	SparkFun COM-09195 28X2
PZ1	1	Piezo transducer	Digi-Key 445-5242-1-ND
SW4	1	Slide switch	Digi-Key 563-1348-ND
IC2	1	Battery manager	Digi-Key MAX1811 ESA+-ND
USB	1	USB-A jack	Digi-Key LUSBA11100-ND
SW1-3	3	Tactile switches	Digi-Key EG1821-ND
C1, C3, C4	3	10 µF tantalum	Digi-Key 718-1199-1-ND
J1	1	3 mm audio jack	Digi-Key CP1-3523N-ND
D88	1	(8x8) dual color matrix	Adafruit ID:458
B1	1	260 mAH LiPo battery	HobbyKing SKU: 9210000070
Standoffs	1	Package of four	RadioShack Cat#: 276-1381
R2-R5	4	10K 1/4 W resistors	Jameco Part#: 2157167
R6	1	470 1/4 W resistor	Jameco Part#: 2159082
R1	1	4K7 1/4W resistor	RadioShack Cat#: 271-1330
R7	1	22K 1/4W resistor	RadioShack Cat#: 271-1339
SOIC to DIP	1	Breakout board	Jameco Part#: 291208
D1	1	3 mm green LED	Jameco Part#: 98950
Circuit Board	1	Prototyping board	Jameco Part#:263207
Optional Parts			
Asst key caps	1	Pkg of 12 asst key caps	SparkFun COM-10302
1,000 mA battery	1	Larger lithium battery	SparkFun PRT-00339
16 MHz XTAL	1	External oscillator	SparkFun COM-0942
28-pin socket	1	28-pin socket for PICAXE	Digi-Key ED3128-ND
Eight-pin socket	1	Eight-pin socket for MAX1811	Digi-Key AE10011-ND

Miscellaneous hardware/tools:
Plastic for faceplate/backplate
Dremel tool
Drill with 1/8 and 1/16" bits
Soldering iron

PARTS LIST



■ **FIGURE 4.** Modification of the plated through hole board.

to make sure that all of the holes line up and that all the parts have proper clearance.

I captured the 29 x 32 pattern of the board using a vector graphics editor. I keep a small library of the images for the parts used in the circuit and replicated the placement using the editor. I use this graphic to plan the interconnections between parts to reduce possible errors. The placement of the components for the top and bottom side of the board along with all the connections is shown in **Figure 6**. These views can be really helpful when it comes time to solder connections to the board.

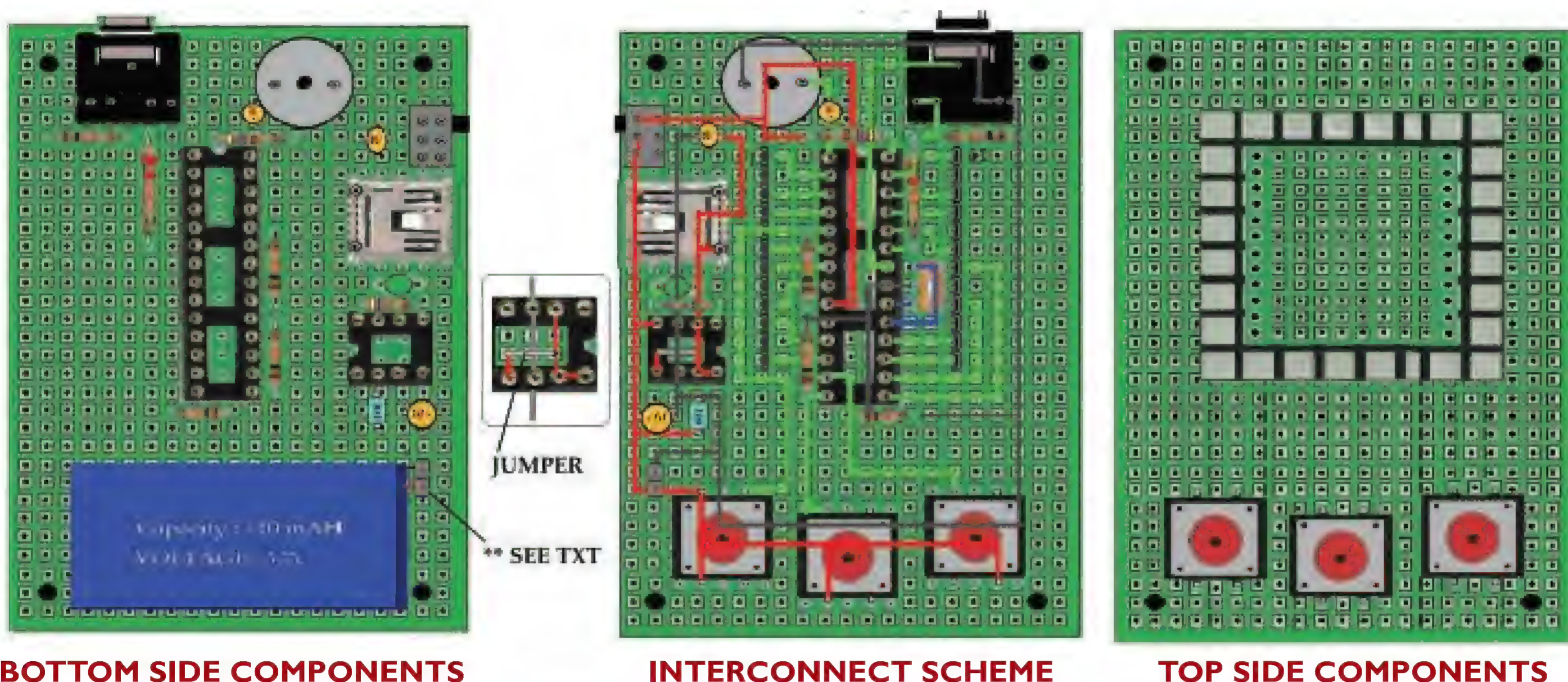
I reflected the placement of the parts to help prevent



■ **FIGURE 5.** Trial assembly of the board using standoffs.

the typical PIN1 orientation error for the two chips used in the circuit. The post shown connected to the battery in the bottom side component view is a two-pin male header. This is something I used as an easy test point and is not required for the circuit to operate. When there are many closely spaced routes on the board, sometimes I find it easier to trace an outline of the route using an ultra fine felt tip pen as a guide. It then just becomes a matter of point-to-point wiring.

There is only one jumper used on the board. It fits

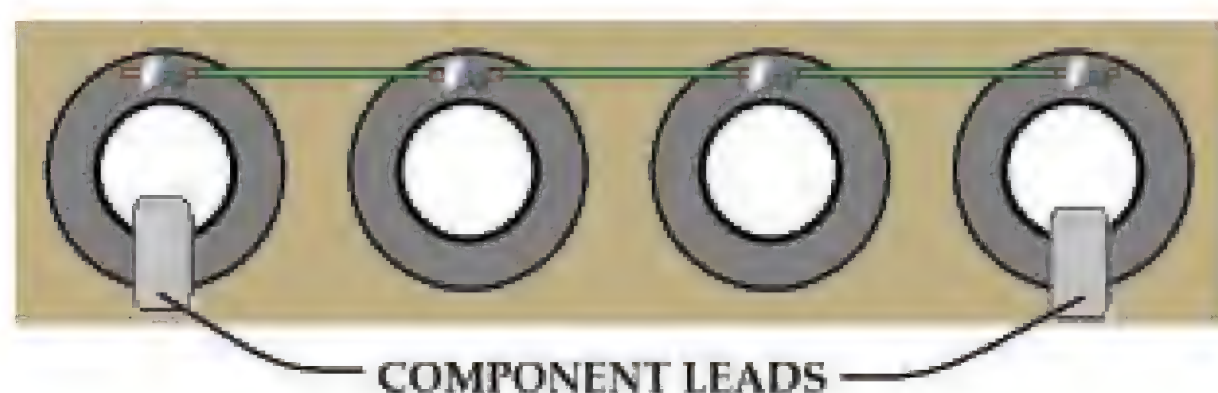


BOTTOM SIDE COMPONENTS

INTERCONNECT SCHEME

TOP SIDE COMPONENTS

■ **FIGURE 6.** Parts placement and interconnect scheme for the circuit board.



■ **FIGURE 7. Interconnect techniques with plated through holes.**

neatly under the eight-pin socket for the MAX1811 and it is shown as a close-up in **Figure 6**. Any interconnect technique should work fine using this plated through hole board. My favorite two methods are shown in **Figure 7**, but use any connection technique you prefer.

The current flowing through the connections is very low so you can use a thin gauge wire that will make it easier to cover up with the faceplate. Keep in mind that with a plated through hole board, the interconnect can happen on either side. I used whichever side was easier to solder and kept most of the connections on the side that had the most components. By using 30 AWG magnet wire, the component leads and wire can fit in the same hole before and after soldering.

To build this circuit, the prototyping board must have plated through holes. The reason for this is because components need to mount on both sides of the board, so a single-side plated board will not work. To cover up the solder points on the top side of the board where the pushbuttons and display are mounted, I used a piece of high gloss black plastic I bought at a hobby store. The problem with this is that you have to drill holes in the plastic to allow the pins from the display and leads from the buttons to feed through.

I used a fine-tip pen to mark the locations on the plastic and used a 1/16" inch drill bit to make the holes. The matrix leads just make it through to the backside of the circuit board for soldering, but the leads on the buttons are too short. This problem can be easily fixed by soldering extensions onto the leads as shown in **Figure 8**. The screws included with the standoffs have built-in lock

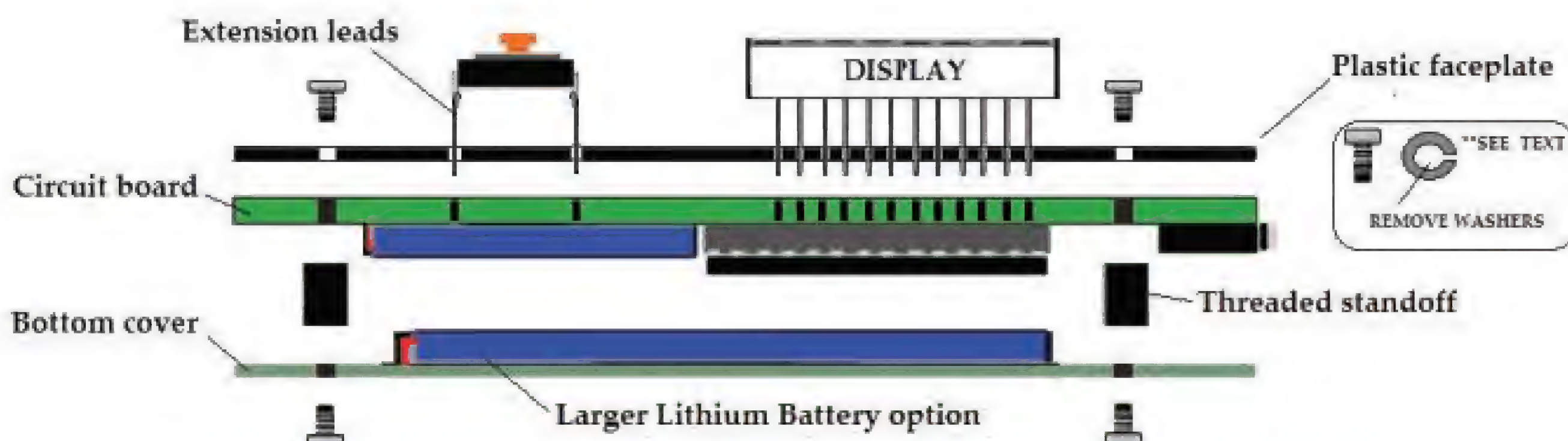
washers that need to be removed to allow the threads to reach through the faceplate. The design needs to have a bottom board; this can be anything you like. I used the same black plastic that was used as a cover plate.

This board performs two tasks. It covers the mounted components and provides a back plate to hold on to when playing the game. Another possible benefit is that it can be used to mount a larger lithium battery if you want a longer period between charges. You can easily fit a 1,000 mA battery in the space between the threaded standoffs. I listed the catalog number for this as optional in the **Parts List**. SparkFun sells a pack of 12 key caps specifically for the tactile switches used for the circuit; they help add a real professional look to the game.

Checking Your Work

Once you have mounted all of the components, it is time to test the circuit. Before you insert the 28x2, turn the slide switch to the on position and test the supply voltage from either ground pin to the +V on pin 20 of the PICAXE. If the supply voltage is fine, insert the processor and connect the programmer to the audio jack. Load the test routine to exercise the buttons, sound, and matrix display.

Once loaded, there should be a blinking square in the center of the display. Depress the left and right buttons, and the square should move from left to right. Press the center button, and the box will extend outward with a changing tone as it expands. This test guarantees a working display, button control, and sound. The code for



■ **FIGURE 8. Parts placement for the top and bottom side of the board.**

this test can be downloaded from the article link, along with the example game; file name is GAME_DIAG.BAS.

How the Game Works

Once you have tested the components of the circuit, you will want to load the example game for the project. The game is entitled DEFLECTO. This BASIC program serves as an example as to how easy it is to program this circuit.

The game begins in the ready to start mode with a red game paddle moving back and forth until you press the center button. After pressing the center button, a ball descends at some random angle toward the bottom of the display as shown in **Figure 9**.

The incoming ball can be deflected by positioning your paddle using the left and right buttons. If you miss the incoming ball, a MISS sound occurs and you must press the center button again to initiate the next ball. If you deflect the ball, it returns upward at an appropriate angle back toward the top of the display.

As the balls get deflected, it turns on the pixel that gets struck as it hits the boundary. If the ball strikes a tile that has already been lit, the ball gets deflected and returns as the ball in play. The game is won after lighting all the tiles across the top.

You may want to rewrite the program so that you must keep the last ball in play until you strike the last tile or you lose all of the seven previously turned on tiles.

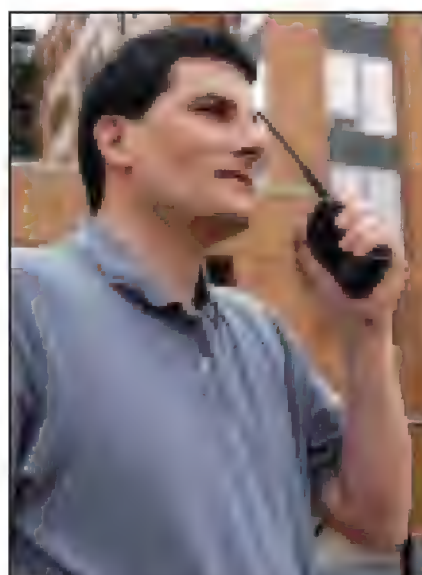
I have broken down some of the key elements of the game design in hopes that it may help you to understand the program better. The specific named ports that tie to the rows and columns of the display have been symbolized to make programming easier. The primary routine in the game is the player paddle position. **Figure 10** illustrates the seven positions the paddle can be in during game play.

The program sets paddle positioning as the primary event, and polls the left and right direction buttons between the start of all other events. A variable used to represent the symbol XPOS is incremented or decremented with each push of the buttons tied to ports C.2 and C.3. If $XPOS = 1$, row R7 and R8 are held low and column C1 is toggled to light the leftmost paddle position. For

■ **FIGURE 9.**
Shot of a
typical game
starting.



the incoming balls, fixed diagonal or vertical lighting subroutines are jumped to randomly whenever the center button connected to the port C.3 is pressed.

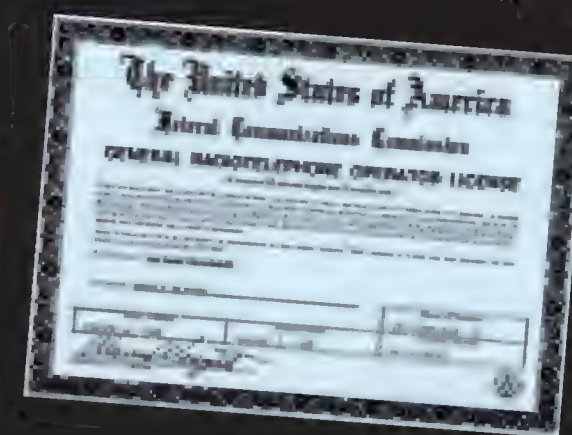


EARN MORE MONEY Get your dream job!

**Make up to \$100 an hour
and more as an FCC
Licensed Technician!**

**Get your "FCC Commercial Radio License" with
our proven Home-Study Course!**

- No costly school. No classes to attend.
- Learn at home. Low cost!
- No experience needed!
- **MONEY BACK GUARANTEE:** You get your FCC License or money refunded!



Turn your interest in electronics into a high paying career in Communications, Radio-TV, Avionics, Marine, Radar, Safety, Municipal and more... even start your own business!

Call for FREE info: 800-932-4268

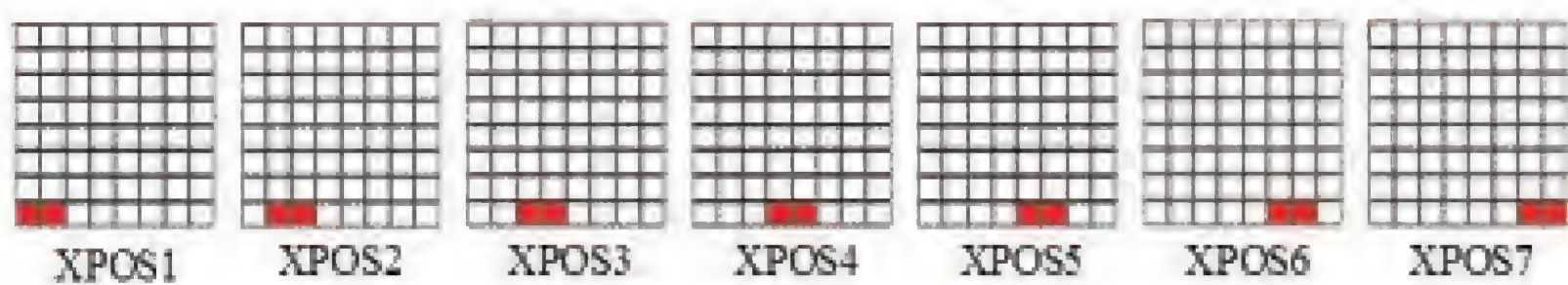
Or, email us: info@LicenseTraining.com ext. 109

www.LicenseTraining.com

COMMAND PRODUCTIONS FCC License Training

Industrial Center, 480 Gate Five Rd., PO Box 3000, Sausalito, CA 94966-3000





■ **FIGURE 10. Example paddle positions.**

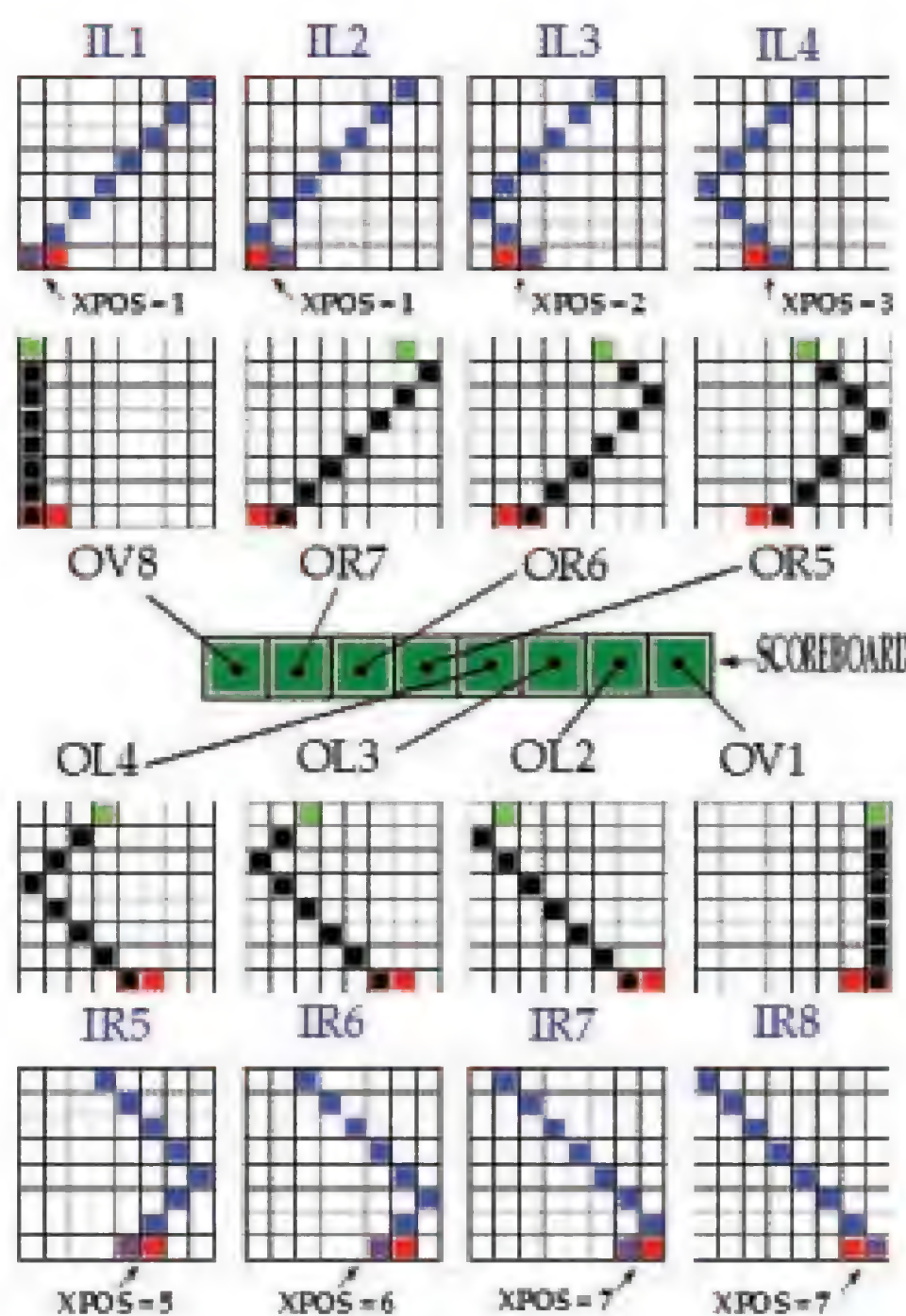
There are eight angled inbound ball routines along with six angled and two vertical outbound returns as shown in **Figure 11**. The game seems very intelligent and complicated when playing, but literally only took about an hour to write.

Deflecting the inbound balls by moving your paddle position to intersect the incoming ball, the rebounded balls should strike all eight upper tiles eventually after successfully returning all eight volleys. Depending on the exact paddle location, the ball returns in a corresponding manner. For example, if the inbound ball IR5 meets a paddle in XPOS = 4, the ball returns vertically.

If the paddle is at XPOS = 5, the ball is returned at an angle. Whenever a deflected ball strikes the top boundary, the individual top LED becomes lit. The game is won after all eight positions

have been hit.

The program contains all of the elements you will need to write your own games, including sound routines, button action routines, position tracking, and more.



■ **FIGURE 11. Lighting the top eight LEDs with the eight outbound balls.**

Game Over ...

Substitutions can be made for most all of the parts in this project. You may have to abandon the exact design if there are great enough differences in size, shape, and lead spacing of the substituted components.

Also, any 24-pin display will work along with any 16-pin models with the appropriate modifications.

I hope that any readers that build this project share their programming efforts through the *Nuts & Volts* forums or at the specific article link. I have several other game ideas that I am willing to share with anyone interested and will make them available as soon as they are finished.

I am working on a printed circuit board for this design and I will make it available to anyone interested. **NV**



PBP3
PICBASIC PRO™ Compiler

Experimenter - \$49.95
Silver Edition - \$119.95
Gold Edition - \$269.95

The industry-standard BASIC compiler for Microchip PIC® microcontrollers.

Multi-Seat Licensing
for Educational Institutions

Upgrade from
PICBASIC™ Compiler (PBC)

Download a FREE trial version now.

www.PBP3.com

microEngineering Labs, Inc.

www.melabs.com

888-316-1753

PICBASIC and PICBASIC PRO are trademarks of Microchip Technology Inc. in the USA and other countries. PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.

Theremin Synthesizer Kit MkII

Create your own eerie science fiction sound effects by simply moving your hand near the antenna. Easy to set up and build. Complete kit contains PCB with overlay, pre-machined case and all specified components.

• PCB: 85 x 145mm
Cat. KC-5475

\$54.00*

POPULAR KIT



Test & Measure Kits

LED Battery Voltage Indicator Kit

This tiny circuit measures just 25mm x 25mm and will provide power indication and low voltage indication using a bi-color LED, and can be used in just about any piece of battery operated equipment. Current consumption is only 3mA at 6V and 8mA at 10V and the circuit is suitable for equipment powered from about 6-30VDC. With a simple circuit change, the bi-color LED will produce a red glow to indicate that the voltage has exceeded a preset value.

- PCB, bi-color LED and all specified electronic components supplied
- PCB: 25 x 25mm

Cat. KA-1778



\$7.25*

USB Power Monitor Kit

Plug this kit inline with a USB device to display the current that is drawn at any given time. Check the total power draw from an unpowered hub and its attached devices or what impact a USB device has on your laptop battery life. Displays current, voltage or power, is auto-ranging and will read as low as a few micro amps and up to over an amp. Kit supplied with double sided, solder masked and screen-printed PCB with SMD components pre soldered, LCD screen, and components.

- PCB: 65 x 36mm

Cat. KC-5516



\$43.25*

Note: Laptop not included

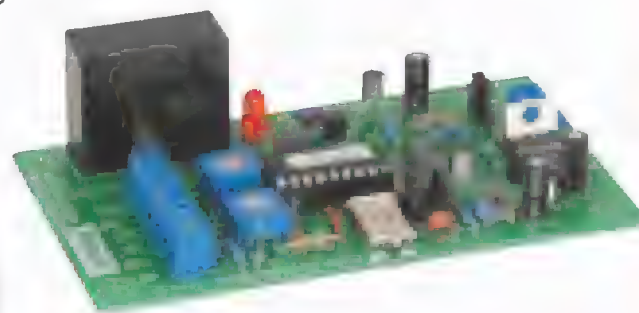
High Performance Timer Kit

This is a very sophisticated timer, which can be adapted for two different types of uses. The first is "one shot" operation, which can be used to keep electric windows active, or a thermo fan running for a period after ignition is switched off etc. The second is a pulse type operation, which can be used to squirt water spray for 1 second every 9 seconds. The uses are endless, and the time is adjustable from 0.1 seconds, to 16.5 minutes via easy to use (and accurate) digital switches. Kit supplied with PCB, and all electronic components.

- PCB: 105 x 60mm

Cat. KC-5379

\$31.00*



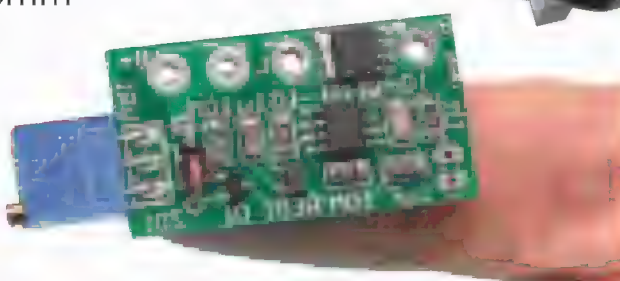
Battery Saver Kit

Cuts off the power between the battery and load when the battery becomes flat to prevent the battery over-discharging and becoming damaged. Suits SLA, Li-ion, Li-Po and LiFePO4 batteries between 6 to 24V. Uses very little power (<5uA) and handles 20A (30A peak). Supplied with double sided, solder masked and screen-printed PCB with SMDs pre-soldered (apart from voltage setting resistors) and components.

- PCB: 34 x 18.5mm

Cat. KC-5523

\$21.75*



NEW 2013

Universal Voltage Switch

A universal module suits a range of different applications. It will trip a relay when a preset voltage is reached. Can be configured to trip with a rising or falling voltage making it suitable for a wide variety of voltage outputting devices e.g. throttle position sensor, air flow sensor, EGO sensor. It also features adjustable hysteresis (the difference between trigger on/off voltage), making it extremely versatile. You could use it to trigger an extra fuel pump under high boost, anti-lag waste gate shutoff, and much more.

- Kit supplied with PCB, and electronic components.
- PCB size: 105 x 60mm

Cat. KC-5377

\$23.75*



POPULAR KIT

Jacob's Ladder High Voltage Display Kit MK2

With this kit and the purchase of a 12V ignition coil (available from auto stores and parts recyclers), create an awesome rising ladder of noisy sparks that emits the distinct smell of ozone. This improved circuit is suited to modern high power ignition coils and will deliver a spectacular visual display. Kit includes PCB, pre-cut wire/ladder and electronic components.

- 12V car battery, 7Ah SLA or > 5A DC power supply required
- PCB: 170 x 76mm

Cat. KC-5445

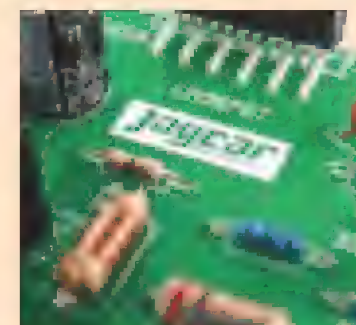
\$31.00*



ATTENTION KIT BUILDERS

Can't find the kit you are looking for? Try the Jaycar Kit Back Catalog

Our central warehouse keeps a quantity of older and slow-moving kits that can no longer be held in stores. A list of kits can be found on our website. Just search for "kit back catalog".



HOW TO ORDER

PHONE: 1800 784 0263*
FAX: +61 2 8832 3118*
EMAIL: techstore@jaycar.com
POST: P.O. Box 7172 Silverwater
DC NSW 1811 Australia

* Australian Eastern Standard Time
(Monday - Friday 6.30am - 5.30pm)

* US Eastern Standard Time
(Monday - Friday 4.30pm - 3.30am)

Prices valid until 31/10/2013

POST & PACKING CHARGES

Order value	Cost
\$25 - \$49.99	\$7.50
\$50 - \$99.99	\$20.00
\$100 - \$199.99	\$40.00
\$200 - \$499.99	\$60.00
\$500+	\$75.00

*ALL PRICES IN USD &
EXCLUDE POSTAGE & PACKING

Max weight: 12lb (5kg)
Heavier parcels: POA
Minimum order: \$25

Please note: Products are despatch from Australia, so local customs duty & taxes may apply.

NOW shipping via



5-6 working days delivery

Don't Just Sit There...Build Something!

Get Amp'ed Up Over the Theory Behind Building a Headset Amplifier

By Ron Anderson
ronwande@bellsouth.net

Last month, I detailed the assembly of a headset amplifier. This month, I will discuss the theory of operation behind that amplifier. However, most of this discussion will apply to the power amplifier I wrote about in the August issue, as well.

PERFORMANCE

Despite the fact my headset project used inexpensive parts, the resulting amplifier (a diagram of which is shown in **Figure 1**) is a high quality device with very low distortion and noise, and a flat frequency response over the audio frequency range and well beyond.

Distortion is also excellent — less than 0.0012% from 20 Hz to 1 kHz, increasing to 0.0021% at 20 kHz as measured with my HP 339A with a 1K load. With a 100 ohm load, distortion increases somewhat to 0.0016% at low frequencies and 0.008% at 20 kHz. At 10 kHz with this load, it measures about 0.005%. The level at 1K load is right at the limit of the sensitivity of this instrument.

Distortion is probably considerably lower than this. Traditional distortion analyzers measure distortion plus noise. THD+N is the common nomenclature: Total Harmonic Distortion plus Noise. Essentially, these instruments tune out the fundamental frequency signal and measure everything else, which includes noise and harmonics. The breadboard circuit produces no audible noise in the headset with no input signal. Measured noise at the amplifier output with a 1K load and a measurement bandwidth of 100 kHz is 20 microvolts.



Headset Amplifier; September 2013 issue



Power Amplifier; August 2013 issue

Go to www.nutsvolts.com/index.php?/magazine/article/october2013_Anderson to post comments.

How good is good enough? It is commonly accepted that 0.1% harmonic distortion is about the limit of audibility. A worse problem is that harmonic distortion indicates non-linearity of the circuit which can produce what is called intermodulation distortion. Low harmonic distortion guarantees low intermodulation distortion. I've tested myself by listening to a pure 500 Hz tone and then gradually injecting a second harmonic until I could just hear it. The distortion analyzer indicated 0.08% harmonic at that point. I feel that it would take considerably more than that to be heard when listening to music.

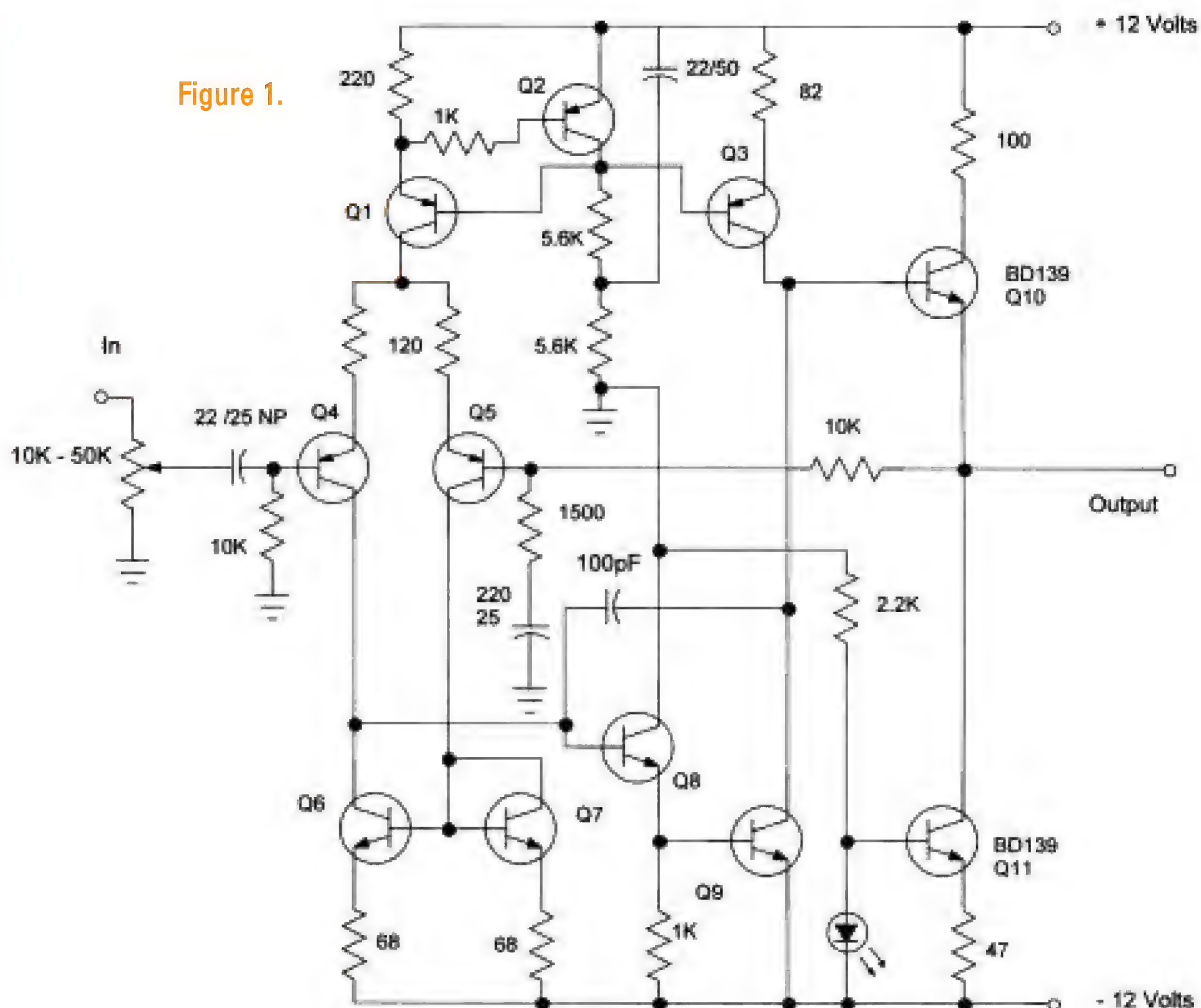
DESIGN

How can such an inexpensive circuit have such impressive performance? The short answer is that the design is much more important than the cost of the components. I take no credit for the design except for the output stage with its inherent shortcircuit protection. The rest of the project is an implementation of an inexpensive amplifier and the verification of its performance.

Since the 1960s, there has been one dominant arrangement or architecture used for most power

- Transistors Q1-5 2N5551 NPN
- Transistors Q6-9 2N5401 PNP
- Output transistors Philips BD139
- Resistors 1/4 watt carbon film
- Aluminum electrolytic capacitors voltages as marked
- 100 pF capacitor must be NPO ceramic or mica

Figure 1.



amplifiers. This design approach was developed by Dr. Hung C. Lin who worked for RCA in the early days of transistors in the area of developing applications. (Google "Dr. Lin RCA" to find interesting articles about him). This has come to be known as the Lin architecture.

There are many variations on this basic design. We have started with (almost) the simplest of them. Because this project incorporates the first two stages of most modern solid-state power amplifiers, the same design can be used to build a low to medium power audio amplifier.

Transistors are not inherently linear devices. Much can be done to make them linear and minimize distortion. A good start for an input stage is to use a pair of transistors in differential mode (Q4 and Q5). A differential input stage reduces the second harmonic to a very low level if it is balanced accurately. The differential stage has two inputs. This provides a very convenient way to connect negative feedback to establish the gain of the amplifier and to reduce distortion.

The pair of transistors in this stage work in opposition to each other. If the signal current goes up in one, it goes down by the same amount in the other. The non-linearities in the two cancel each other. The extent of the cancellation depends on the balance of the two. The 120 ohm emitter resistors reduce the gain of the stage somewhat and also reduce the distortion. They make the matching of the two transistors less critical.

Transistors Q1-Q3 at the top of the **diagram** comprise a voltage reference and two current sources. Q2 is the reference. It controls the emitter voltage on Q1 and Q3. The current is adjusted by the value of the emitter resistor in the current source transistor. The input differential amplifier's current source is about 3 mA, set by the 220 ohm emitter resistor on Q1.

Using a current source provides a reduction in distortion when compared to using a resistor. It also greatly improves the "power supply rejection" – the sensitivity to power supply ripple. For pennies, the improvement in the circuit it provides makes it very much worth including.

The voltage reference is such that about 0.64 volts appear across the 220 ohm resistor – a current of about

3 mA. This source provides what is commonly called the "tail current" and the input stage is called a "long tailed pair." An ideal current source would provide absolutely constant current regardless of the load resistance over a wide range. This simple circuit comes close.

Balance is critical so the load for the differential stage is a "current mirror" (Q6 and Q7). It provides an active load for the transistor and it balances the current between the two transistors at zero signal, so the current is very nearly the same in each. The current mirror has emitter resistors of 68 ohms which compensate for small differences in V_{be} and beta. The addition of these two inexpensive transistors and a pair of resistors can improve the distortion level by a factor of 100.

Incidentally, I've built a number of amplifiers using this input stage configuration and have never matched the transistor pairs. The circuit is more or less immune to variations in transistor beta and V_{be} .

This first stage drives the base of Q8 which has its emitter connected to the base of Q9. This arrangement greatly enhances the current gain of this second stage. The first stage is a "transconductance" stage which simply means that it converts the input voltage to a current. The second stage converts that current to a signal voltage. We want a large voltage gain in this stage, so we use a current source as the load.

This source is about 8 mA provided by Q3. It uses the same reference voltage as the first stage's current source. The 82 ohm emitter resistor sets the current. The AC impedance of the current source is greater than half a megohm, which makes the voltage gain high. This second stage is called the voltage amplifier stage, or VAS for short.

In this simple amplifier, the output of the VAS drives

an emitter follower output device. Q11 is a current source which acts as a load for the emitter follower Q10. (Yes, I know by convention this is a current sink.) The green LED acts as a voltage reference of about 2.1 volts. The 2.2K ohm resistor supplies about 5 mA to turn it on. The reference voltage is applied to the base of the current source transistor. The emitter resistor of the current source is 47 ohms, which sets the current at about 35 mA. The exact value is not critical.

The current from the positive supply through Q10 is limited by the 100 ohm resistor in the collector. The power dissipation in the output transistors is about 0.35 watts in each with no signal. The output transistors could operate without the heatsinks and be well within their ratings, but I like to play it safe. This is a Class A output stage.

The current source active load is better than a resistor because it can supply full current to the output even when the signal voltage nears the negative supply voltage. The 100 ohm resistor in series with the positive supply protects the amplifier output from accidental shortcircuits of a short duration, i.e., a second or two. In the case of a long term short, this resistor will become a fuse!

In order to keep the circuit from becoming an oscillator, there is a compensating capacitor from the collector of the VAS to the base of the input follower transistor. Ordinary ceramic capacitors are notorious for changing value as the applied voltage changes. If you use a ceramic here, it must be of the NPO type (indicating that the voltage coefficient of capacitance is around zero, but may be slightly negative or positive). A mica capacitor is a little more expensive but guaranteed to work well.

The various transistors in the signal path each have an upper frequency limit. That is, they act like a low pass

filter. Each additional transistor in the signal path contributes phase shift at some high frequency. This phase shift is cumulative and if it reaches the point where the negative feedback becomes positive, the result will be a peak in the high frequency response. A little more phase shift produces an oscillator at some high frequency.

The purpose of the compensating capacitor is to provide a low pass filter much lower than any of the others, so that the frequency response is reduced as frequency increases. As long as the "open loop" gain of the amplifier reaches unity as frequency increases before the phase shift exceeds 90 degrees by "too much," the amplifier will be stable when negative feedback is applied. "Too much" is rather ambiguous. That is a long story beyond the scope of this article. (If you are interested in pursuing this topic, look up "phase margin" in any of the books listed in the **sidebar**.)

Notice that a 10K resistor is connected to the output that goes back to the negative input of the first stage. This is the negative feedback path. This resistor and the 1,500 ohm resistor to the 220 μ F capacitor form a voltage divider that sets the amount of negative feedback, and thus the AC gain of the amplifier — about 7.7. The 220 μ F capacitor allows full DC voltage feedback, so the output DC voltage can be on the order of a few millivolts.

(The voltage on the 220 μ F capacitor will be positive a few tenths of a volt in normal operation, so it need not be the non-polar type). The amplifier's voltage gain at DC is unity. Note that the signal gain of the amplifier without feedback is VERY high. The amplifier won't work without the feedback connected because it is the feedback that puts the operating point of the voltage amplifier stage such that the output is near zero with no signal. (Whew!)

What we have built here is a discrete component

REFERENCES

High-Power Audio Amplifier Construction Manual by G. Randy Slone, McGraw Hill 1999

This one is useful as a project book. Many of the designs are suitable for building. It is much more than a construction manual, however. Slone discusses the most prevalent designs in audio amplifiers. Not a great deal has changed in the years since the writing of this book, though some of the transistors used in these designs are either hard to find or expensive due to their scarcity. They are still available if you look hard enough. Slone discusses subjectivity briefly.

The Audiophile's Project Sourcebook by G. Randy Slone. TAB division of McGraw Hill 2002

This book has a number of audio projects of various complexity including stepped attenuators, various preamplifiers and tone control circuits, electronic crossover networks, power amplifiers, an electronic reverberation circuit, and many more. This one *is* a cookbook.

Audio Power Amplifier Design Handbook by Doug Self, Focal Press 5th Edition 2009

This is the most technical of the books, containing lots of data interesting to an engineer. It covers a number of design possibilities. Self enumerates the causes of distortion in an amplifier and systematically shows how to minimize each of them, resulting in

what he calls the "blameless" amplifier. It is not "perfect," but all the known sources of distortion have been considered and adjusted to minimize the distortion. This book has the best discussion of subjectivity. An older version (i.e., the fourth edition) may still be available. There is a LOT of new material in the fifth edition.

Designing Audio Power Amplifiers by Bob Cordell, McGraw Hill 2011

Bob has the ability to describe much of the technical detail in a way that is more intuitive than most authors. It is not a cookbook, but it is a good discussion of the subject. The circuits in this book use transistors that are newer and more available (and less expensive and a bit less capable) than those in Slone's book.

All of the above books discuss the (now almost) standard three-stage Lin architecture and numerous variations for each stage. Self describes some of these improvements in detail and concludes that though the "numbers" are better, the improvement is not audible. Perfection is a worthy goal, however, never to be achieved in an audio power amplifier, but some designs come very close. I purchased all four of these books at **Amazon.com**. I'm not sure all are still available there.

What is subjectivity and the subjectivist movement? Follow this link for some interesting reading. You don't have to understand all of it to appreciate what Doug Self has to say at **www.douglas-self.com/ampins/pseudo/subjectv.htm**. The discussion is almost word for word from a section of his book mentioned above.

operational amplifier capable of a little more power output than most of the integrated circuit ones.

HEADSETS

Headset impedance varies from 600 ohms down to 24 ohms — possibly beyond this range. The volume control at the input will allow almost any signal source to drive the amplifier to overload, so turn it up from zero until the headset signal is sufficiently loud.

An amplifier on perfboard laying on a bench is an accident waiting to happen. Mount your board in a project box on some standoffs. Of course, you will want the box to be large enough for two channels. Place the power transformer as far from the input connections as you can.

I've tested this amplifier with the following headphones: a Sennheiser HD 202 (32 ohms, approx. \$30); a Sennheiser HD 650 (330 ohms, \$400-\$500); and with a AKG 702 (60 ohms, \$350-\$539). The AKG seems the least sensitive. I prefer the HD 202 to the AKG. The HD 650 is great!

DAMPING

Audio folks discuss the damping factor as important to a loudspeaker. Damping factor is the ratio of the speaker or headset impedance to the amplifier output impedance.

References online indicate that for a headset, a damping factor of eight or more is adequate. This amplifier's output impedance is less than 0.05 ohms, so the damping factors are quite high.

I've measured the impedance vs. frequency of the three headsets I just listed. All have a slight peak at about 50 Hz. One set measured about 15% higher impedance at the peak, and a better one was about 10%. A low damping factor would result in the bass response being up about 1.5 dB at 50 Hz in one case, and 1 dB in the other.

This is a small difference but probably detectable by someone with good ears. That is, with a low damping factor this might be audible.

A high damping factor greatly

reduces the peak in the response.

In this design, I agonized over damping factor vs. protection of the amplifier and headset. Initially, I had the 100 ohm resistor in series with the output which provides current limiting, but a terrible damping factor for low impedance headsets. Putting the resistor in the collector circuit of the output follower gives the same current-limiting protection but the negative feedback from the output decreases the output impedance (as long as the amplifier is not overloaded) to a fraction of an ohm. Hopefully, this discussion will help when you design your own projects. **NV**

ALL ELECTRONICS CORPORATION

DUAL PORT USB CAR CHARGER, 2.1 AMP

Creates two USB charging ports in vehicle, 2.1 Amp capacity. For phones, iPads®, Kindle and other portable devices. **CAT# MT-802**



\$7.50
each

DPDT ON-ON MINI TOGGLE SWITCH

RATED: 6 amp @ 120 Vac. Solder lugs. 0.43" long handle. **CAT# MTS-8**

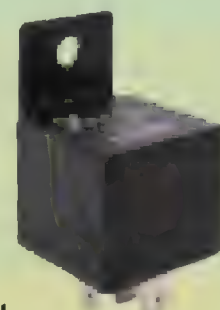


\$1.75
each

10 for \$1.60 each
100 for \$1.35 each

12VDC SPDT 40A AUTOMOTIVE RELAY

12 Vdc, 88 ohm coil. Contacts rated 40A. Plastic flange for bulkhead mounting. Mounts in standard automotive relay socket. **CAT# RLY-351**



\$2.55
each

10 for \$2.25 each

SELF-FUSING RUBBER TAPE

Stretch & wrap self fusing rubber tape for a moisture-tight, insulating seal on electrical connections. 1" x 16.4' roll. **CAT# SFT-5**



\$4.50
each

DOOR LOCK ACTUATOR

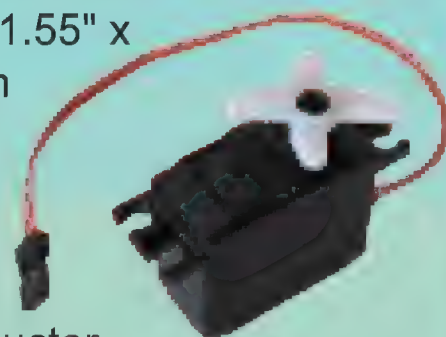
12 Vdc motor-drive actuator for automotive door locks. Nylon plunger moves 0.75". Push or pull, depending on polarity. Overall dimensions (retracted): 5.50" x 2.45" x 1.25". Rubber boot protects against moisture and dust. **CAT# DLA-1**



\$5.50
each

STANDARD RACING SERVO

Analog servo motor. 1.55" x 0.73" x 1.46". 49oz-in torque @ 6Vdc. Speed: 0.23sec/60° @ 4.8V, 0.19 sec/60° @ 6V. 6.5" leads with standard 3-conductor female connector. **CAT# DCS-110**

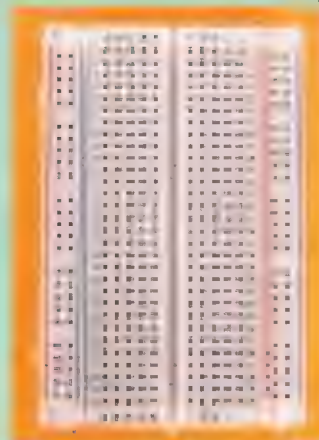


\$6.95
each

5 for \$6.75 each

SOLDERLESS BREADBOARD, 400 CONTACTS

Design and change circuits quickly. Accommodate all sizes of dips and discrete components. Interconnect with solid hook-up wire (#22 AWG recommended). Wires and components can be re-used many times without damage to board or components. 300 contacts on main board and 50 contacts on each of two removable power distribution strips. **CAT# PB-400**



\$4.00
each

2-CONDUCTOR WATER-PROOF CONNECTORS

Mating, polarized connectors. 12" red and black 18 gauge stranded wire leads. **CAT# CON-320**



\$1.85
each

10 for \$1.70 each
100 for \$1.25 each

COLOR HEAT SHRINK TUBING ASSORTMENT, 154 PCS

4 inch lengths of 6 different diameters:
1/16" - 56 pcs
3/32" - 35 pcs
1/8" - 28 pcs
3/16" - 18 pcs
1/4" - 9 pcs
3/8" - 8 pcs
CAT# HS-4900



\$11.95
each

ORDER TOLL FREE 1-800-826-5432
www.allelectronics.com

MakerPlot ^{part 1}

The DIY Software Kit

By John Gavlik and
Martin Hebel

Go to www.nutsvolts.com/index.php?/magazine/article/october2013_MakerPlot for any additional files and/or downloads associated with this article. You can also post comments.

Since you're a *Nuts & Volts* reader, you probably have done — or will do — a microcontroller project at some point. When you do, you're probably going to generate data with your micro. So, have you thought about how that data is displayed, logged, debugged, or otherwise gets presented to you for display and analysis?

MakerPlot is available as a FREE 30 day trial download from www.makerplot.com. If you like what you see and what it does, you can order it from the NV Webstore at a discounted price!

With the popularity of the Arduino, PICAXE, Propeller, Raspberry Pi, BeagleBone, and other micros now on the market (with some of them prominently featured in this magazine), the choices for the hobbyist have no real bounds these days. Given all these great microcontrollers, building a cool circuit and programming one of them is a great first step.

When things go wrong, however — especially in the firmware area where it then becomes a show stopper — what mechanisms do you have to debug your code? One way, of course, is to examine your code line by line and “hope” that you’ll discover the error in your ways. It’s worked for us in the past as we’re sure it has for you, but it’s time-consuming and iffy at best.

Once you find the errant instruction(s), you can re-compile the code and test it again (and again, and again) until it works. This takes a lot of discipline and a good “feel” for what your code is supposed to be doing. It’s doable, but it’s not the most efficient in terms of time spent debugging. As a result, it may take lots of head scratching — not to mention many compiles — to solve the problem. Using this method, you don’t have any visibility into what’s going on inside your micro; your thought process is all you have.

To Plot or Not To Plot

Another way to debug your code is to output the

analog or digital data that your code generates to your monitor in order to get a visual image of what’s happening. Terminal programs like HyperTerminal can do this, but the data are all numbers. This makes a lot more sense compared to just “visual” code parsing as it can lend some hints to what’s wrong, but numbers alone don’t tell the whole story of what’s going on that’s causing your problems. Instead, a picture is worth a thousand words (or numbers).

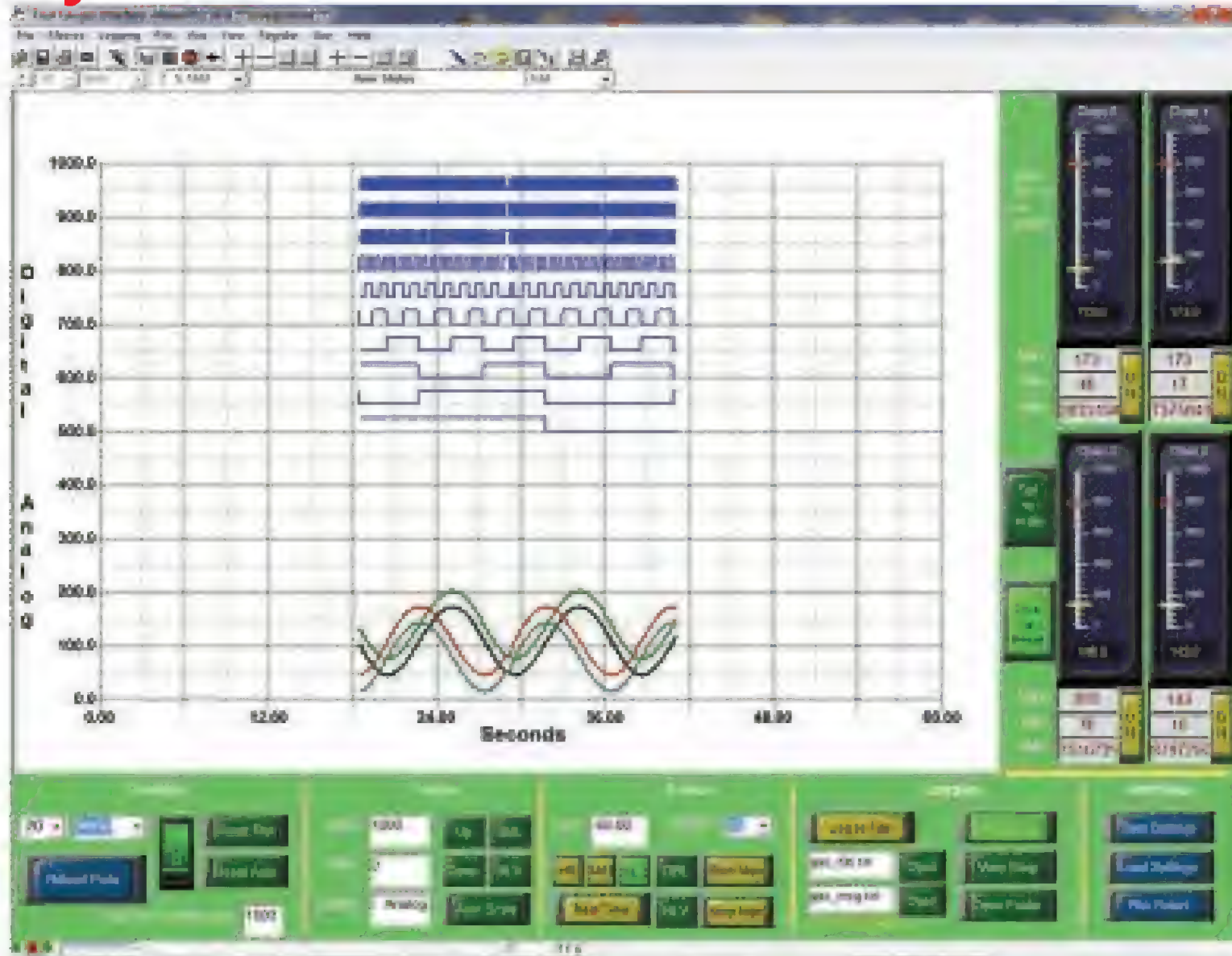
So, if the internal data you’re displaying is graphic in nature (like plot lines of individual data), it’s even simpler. With this graphical information, you can locate your problems much faster and with greater precision as compared with just number-based debugging or manually stepping through your code to try and find the problem.

This is especially true if you’re trying to isolate intermittent problems that occur infrequently, like those that depend on an external input voltage source — a potentiometer or light sensor value that’s converted by an A2D converter into a digital value (threshold and alarm settings come to mind), for example. Let your code run normally, adjust the pot and light levels, and the graphic display will probably show you where the problem lies instantly.

Enter MakerPlot

Unless you’re a purist that nearly always builds their own handmade circuits, you’re generally going to purchase an off-the-shelf “working” hardware board with your favorite micro embedded into it. Since you didn’t design and build the hardware portion yourself, you can (correctly) assume the hardware part of your project is working given that you’ve applied the correct wire hookups, voltage inputs, etc. All these new micro boards are great hardware animals, and they all generate analog and digital data. So, the questions remain: “How do you plot, log, and display this data?” and “How do you debug your code without an external reference as to what’s going on?”

Figure 1.



While you can “roll your own” graphics program to do debugging, that takes a lot of time, skill, and effort. There are a few software programs out there that can do this, but most are either too fundamental or too over-the-top expensive and hard to use. Free, or otherwise, they just don’t do the job. Lucky for you, there’s MakerPlot (**Figure 1**). MakerPlot is Windows software for plotting, data logging, displaying, and debugging analog and digital data generated by your microcontroller. The best part is that there is no proprietary hardware required to get the

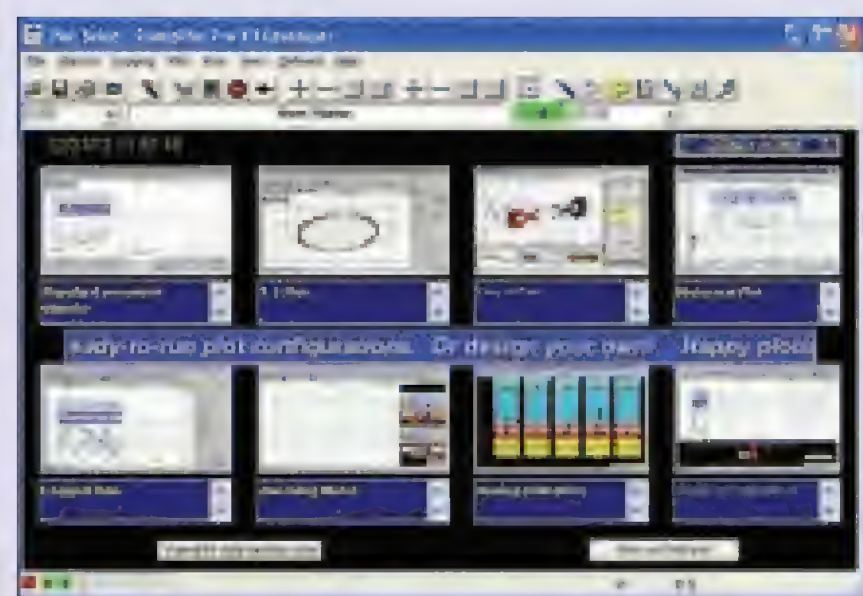
StampPlot to MakerPlot

If you’ve ever used the Parallax BASIC Stamp, you may remember a software program called StampPlot. MakerPlot is the successor to StampPlot. Like StampPlot, MakerPlot is authored by Martin Hebel — Associate Professor at Southern Illinois University and also author of several Parallax publications including *Process Control*, *Industrial Control*, and (along with yours truly, John Gavlik) *Experiments with Alternative Energy*.

With the maker movement in full force, StampPlot was totally re-engineered and renamed to MakerPlot to reflect the direction of this world-wide influence. Many months of coding and debugging went into the revision of StampPlot to MakerPlot — not to mention better looking and more functional graphical controls like switches, meters, text boxes, LEDs, and so forth.

MakerPlot supports just about any microcontroller out there

that can output analog and digital data in a serial ASCII format. So, if you remember StampPlot (or even if you don’t), you’ll be even more pleased with MakerPlot and its expanded capabilities and documentation.



MAKERplot

Data Flow

Figure 2.



data onto your PC monitor — just a serial connection from your microcontroller to the computer. That’s it!

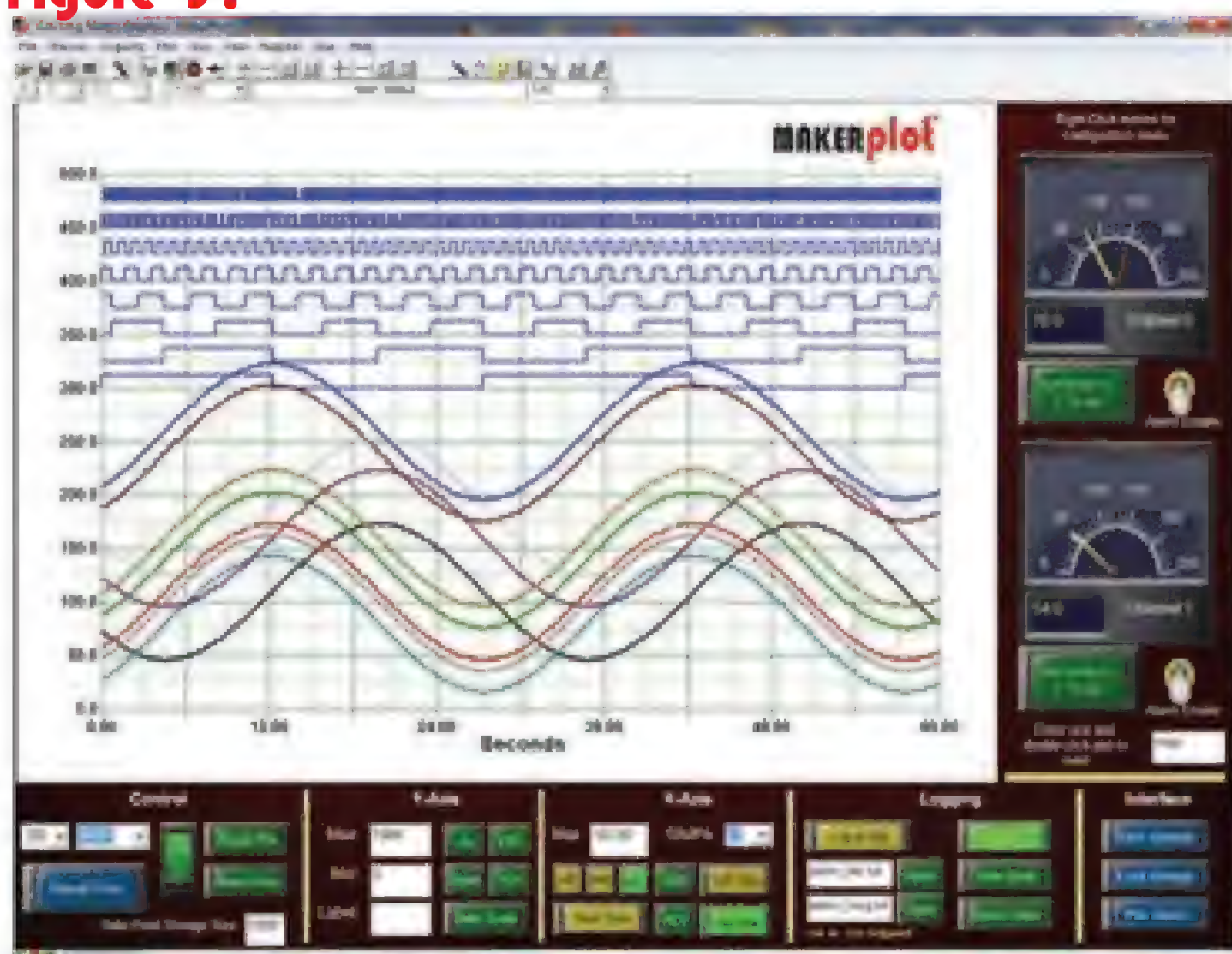
Something as simple as a two-wire RS-232 hookup (TX and Gnd) between your micro and your PC’s serial port is enough. A lot of modern PCs don’t have nine-pin D connectors anymore, so a USB connection will also work. As you well know, this is standard these days with the PC establishing a virtual comm port for the serial data from the micro. This way, you can have fast two-way serial communications between the PC and your micro — something that MakerPlot fully supports. Your code debugging just went to a higher level!

Figure 2 is a functional block diagram of MakerPlot’s data flow. You can see the various paths that your data can take going in and out of the MakerPlot software engine, and there is ample capability for true bi-directional interaction between MakerPlot and your micro. Besides plotting analog and digital data, think of MakerPlot as an extension to your microcontroller for data logging, debugging, and math co-processing. Yes — full blown, floating point math co-processing for your integer-based micro is possible using MakerPlot (more about this later in a subsequent article).

For now, let’s look at how MakerPlot can debug your micro’s analog and digital data. Besides data logging in Excel-compatible CSV format, you can replay data right on your PC’s monitor in a continuous loop so that you can isolate your micro’s firmware or hardware problem. You can also capture the screen as a jpg image and save it to a file for a permanent record of what happened. And, of course, you can print out the data for a hardcopy.

Besides a hard-wired serial connection from your micro

Figure 3.



to your PC, you can also monitor your microcontroller remotely via the TCP/IP protocol built into MakerPlot. So, no matter where your micro is physically, you should be able to access its data using MakerPlot. This makes for an excellent debugging tool, let alone a powerful way to monitor remote data even when everything is working.

An Example of MakerPlot's Capabilities

As an example of what MakerPlot can do, let's assume you have a microcontroller project that involves plotting both analog and digital data. You can do this in

```
#include <SPI.h>

// Analog and Digital Example
// Plots 2 analog channels and 8 bits of digital data

int x; // initialize variable
int val0;
int val1;

void setup()
{
    // configure hardware
    // initialize serial communications at 9600 bps
    Serial.begin(9600);
}

void loop()
{
    for (x= 0; x < 255; x++)
    {
        val0 = x;
        val1 = val0 + 20;

        Serial.print(val0); // plot val0 and val1 analog data
        Serial.print(","); // as saw-tooth plots
        Serial.println(val1);

        Serial.print("[]"); // configure for digital data []
        Serial.print(x); // use "x" as variable
        Serial.println(" RDC 8"); // plot 8 digital channels

        delay(50); // delay 50 milliseconds
    }
}
```

Figure 4.

near real time as shown in **Figure 3**. Here, you see 10 channels of analog data and eight bits of digital data being plotted along with two meters that display any of the two analog channels that you choose. You can also configure meter alarms to alert you when your analog signals are above or below limits that you set. The bottom menu buttons control the horizontal and vertical scales, as well as data logging and screenshots for recording.

Connecting to MakerPlot

MakerPlot is data acquisition and graphical plotting software that runs on Windows computers. You can also run it on a MAC, but you'll need a suitable emulation program. So, "How do you get your micro's data into MakerPlot?" You do it through your micro's serial port. Once again, modern PCs don't have nine-pin D connectors anymore, so you'll probably need to connect your micro to your PC with a USB cable and driver software on the PC side that can create a virtual comm port.

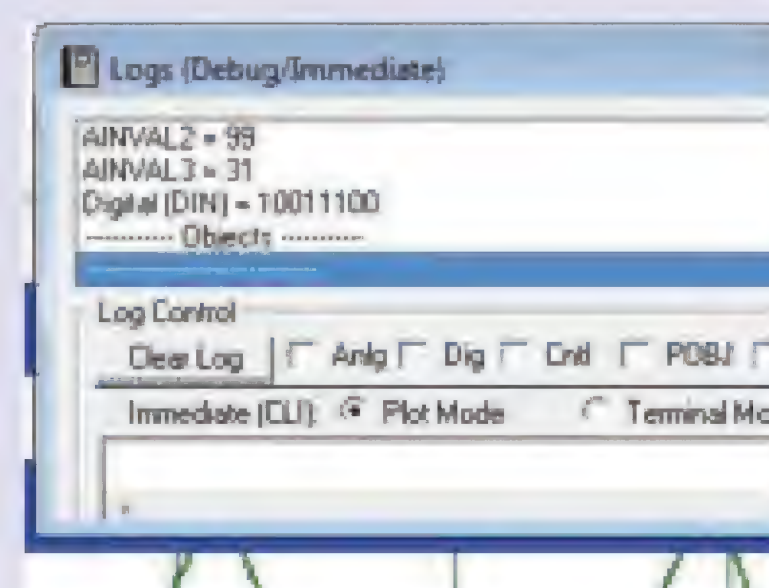
While we like many of the popular micros out there, the Arduino happened to be the most popular at the time of this writing, so most of our code examples are oriented to it. **Figure 4** shows an Arduino sketch that generates analog and digital data for testing and — at the same time — shows how easy it is to create a serial connection (just look at the setup). If you don't use an Arduino, you can take the methods from this example to do your own firmware code using the micro of your choice.

The sketch defines the three variables with the serial comm port at 9600 baud. Then, it goes on to plot two analog data channels as *val0* and *val1*, followed by plotting eight channels of digital data from 0 to 255 and repeating. The analog data are simply sent as comma-separated decimal values ending with a carriage return.

The digital data is really the "x" value used in the *For Loop* that is then converted to eight digital "1" and "0"

The Logs (Debug/Immediate) Window

As this article points out, MakerPlot has a lot of debugging capabilities built into it. That is no more evident than in the Logs (Debug/Immediate) feature. With it, you can witness every analog and digital data point, as well as messages and other MakerPlot real time data streams. Plus, you can manually input analog and digital data (and commands) into MakerPlot to test your code. Everything you do is logged and displayed to you in real time, so you have a record of what you did. For more information, check out the video tutorial series at www.makerplot.com.



values. The MakerPlot “%” prefix command informs the software that the data is digital (binary), and by using the `[value ADC 8]` string, MakerPlot converts the decimal value of x to eight bits of binary. The important thing to realize is that both analog and digital data are being generated as ASCII characters going to MakerPlot. You can see this by looking at the `Serial.print` and `Serial.println` instructions.

Figure 5 is what it looks like with the code running on an Arduino Uno shield. (By the way, we call the MakerPlot display screens “Interfaces.”) So, if you’re interested in plotting, logging, debugging, and otherwise displaying your micro’s analog and digital data in graphical form, MakerPlot is the way to do it.

Why the Name MakerPlot??

We named it MakerPlot for those “makers” out there who want to create simple yet powerful screen interfaces for their projects and products. As you can see, MakerPlot looks like the face of a conventional instrument. With dozens of meters, buttons, switches, and text boxes to choose from, you have full control of how the screen interface is designed and how information is displayed, plotted, logged to files, printed, and more. Your microcontroller also has the ability to read information directly from these same controls for interactive measurement, plotting, and control.

For example, if you have a slider control on the MakerPlot Interface screen, your micro can read the slider value and react accordingly. This makes for the start of some great bi-directional project ideas (more to come on this subject in a subsequent article).

MakerPlot is Customizable

With all this capability, where does one start to figure out what can be done with arranging these controls on the Interface screen? **Figure 6** shows just four off-the-shelf Interfaces that come with MakerPlot. Ten of these Interfaces come standard, and you can build ones like these on your own because you’re provided with all the instructions to do so.

The ability to design a customized graphical user interface (GUI) for monitoring your data — combined with the ability to program your microcontroller to interface directly and simply with the GUI — is a powerful combination for a developer. That’s the real power behind MakerPlot. It’s designed to be customized — by you!

The Plot Thickens ...

While we started this article with asking how you

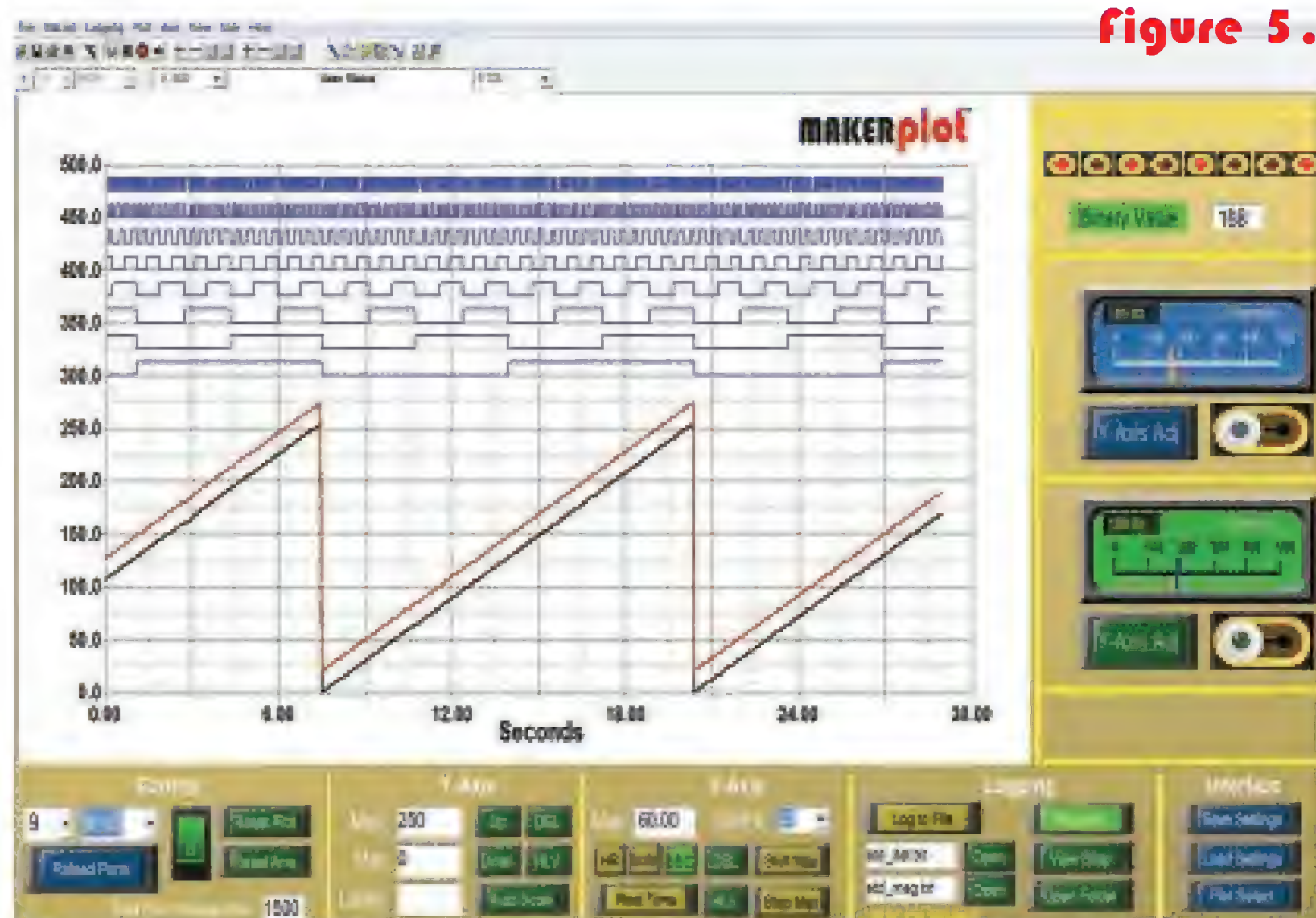


Figure 5.

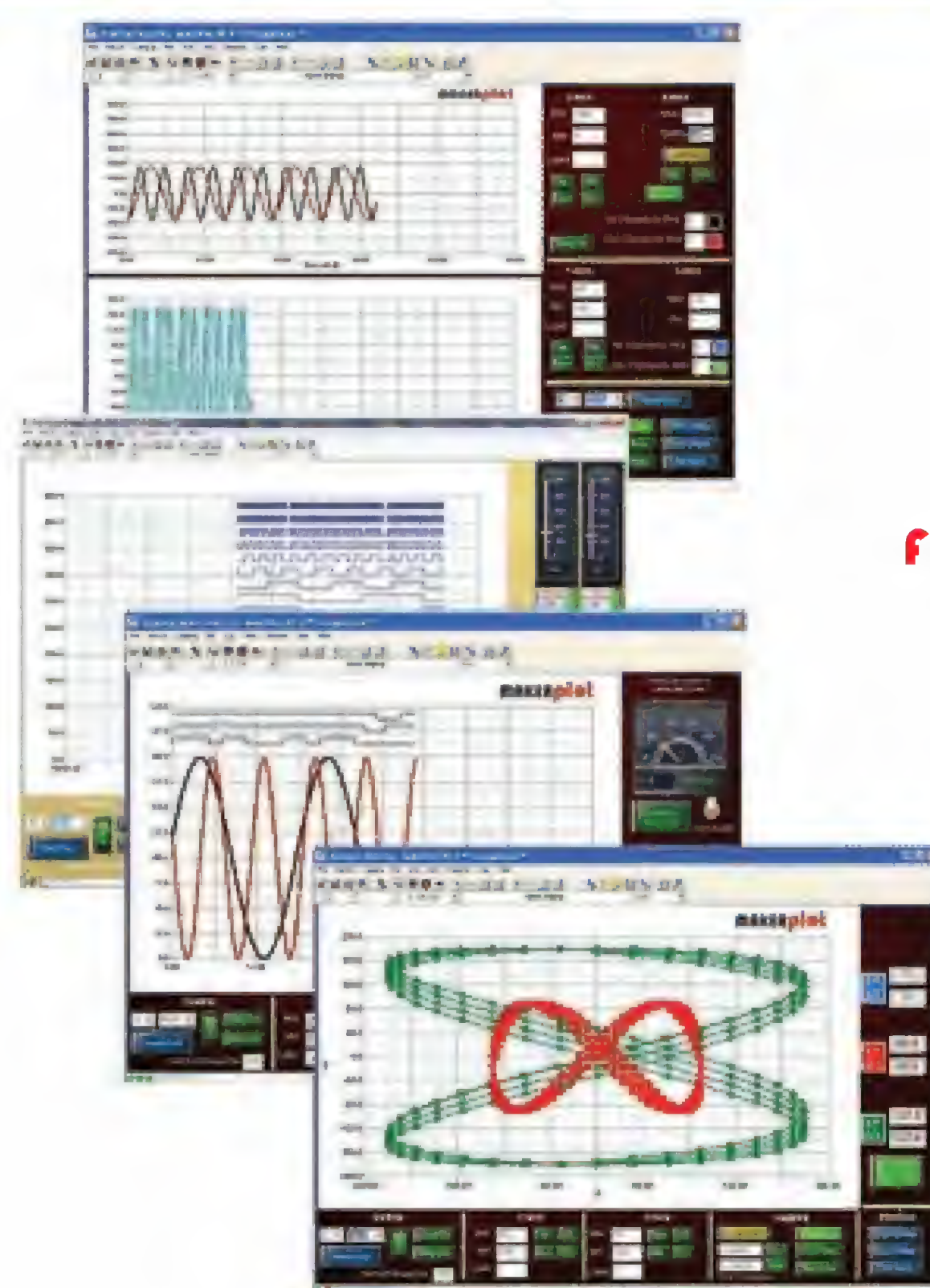


Figure 6.

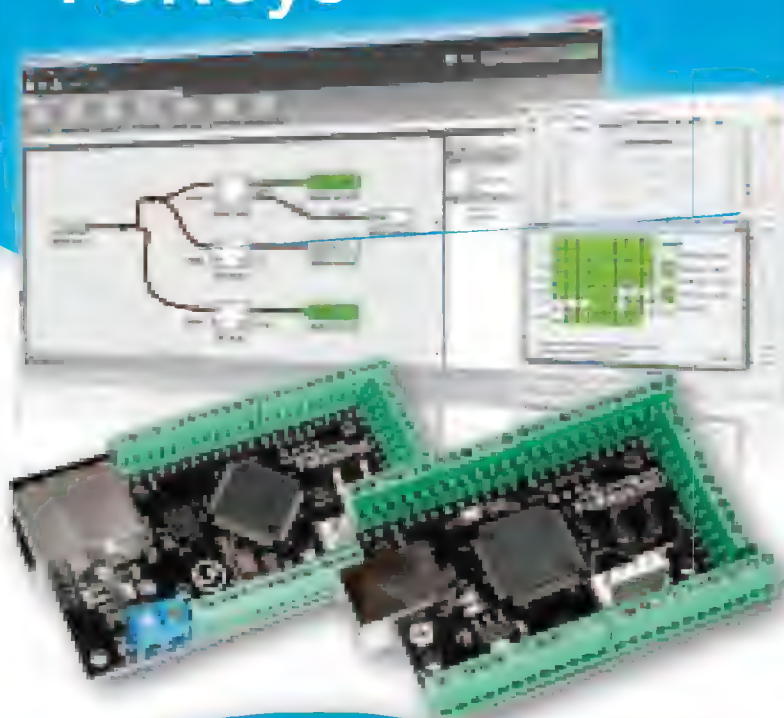
would debug your micro’s data, the discussion of this topic just naturally got expanded into a greater universe of MakerPlot’s graphic capabilities.

We’ll present much more about MakerPlot in subsequent articles — especially on the topic of how you can customize it. If you don’t want to wait, full details are available at www.makerplot.com, including more sketch examples, video tutorials, and a complete MakerPlot guide to show you all the inner-workings of this unique software program. That’s all for now, so just remember: Got data? MakerPlot it! **NV**

Imagine this...

- a small and inexpensive
- USB or ethernet device
- with a rich set of features
- including PLC functionality and 3-axis stepper motor controller
- all accessible via free .NET, ActiveX or C++ library
- cross-platform
- configurable with free software

PoKeys



Or this...

- Smallest USB 2.0 portable 1MS/s oscilloscope
- Data acquisition of analog and digital signals
- Data recording
- Export to CSV, XLS, PDF and HTML
- Simple usage of advanced features
- Examples for C++, VB, Delphi and LabView
- Free software and updates

PoScope Mega1+



POSCOPE
Visit us at
www.poscope.com



READER FEEDBACK

Continued from page 7

manufacturer who has the aqueous system to clean it off the boards. For us home shop workers and hobbyists, our best choices are the old fashioned rosin-based fluxes such as Kester 44 or one of the "no wash" solders that are available.

Al Hobbs

Good Timing

I have a question about a circuit in "Herding Data Over Bridges" in the February 2013 Design Cycle. In Schematics 3 and 4, Fred Eady details a PIC18F46J13, and there is a separate crystal (X2, 32.768 kHz) to drive the internal RTCC. The PICs I use in my circuits (PIC16F887) are much more primitive, but when using Timer1 to provide an accurate timekeeping function, I have found that linking it to the 4 MHz main oscillator crystal is just as accurate as using a separate 32.768 kHz crystal. One clock I designed only gained one second over 12 months (inside my very climate-controlled house)! Why do you include the second crystal? Does the RTCC of that PIC not have any option to run off of the main oscillator?

Thank you.

Judy May W1ORO
Union, KY

Hi Judy! Thanks for reading Design Cycle.

The RTCC component of the PIC18F46J13 was designed to run independent of the core processor. That means the RTCC can continue to run and keep accurate time, even if the microcontroller is sleeping.

To be able to do this, the RTCC clock source must be a stand-alone entity.

The PIC18F46J13's RTCC can be clocked at 32.768 kHz by placing a 32.768 kHz crystal across the Timer1 T1OSCO and T1OSCI pins.

The RTCC can also be clocked by an internal RC oscillator (INTRC). The RTCC clock source is selected by the

RTCOSC configuration bit CONFIG3L<1>. There is no provision to clock the RTCC from the CPU clock.

When the INTRC clocks the RTCC, the tolerances of the internal RC oscillator determine the stability of the RTCC timing. The INTRC clock frequency is stated to be around 31 kHz. You will find that the crystal clock option is a better choice for accurate long-term timekeeping. To assure maximum timing accuracy in crystal mode, a calibration register (RTCCAL) is provided for fine tuning and can adjust the RTCC timing up to ± 2.64 seconds per month.

Fred Eady

PICing My Brain

I have a question about Thomas Henry's circuit in "PIC Trainer from Surplus Parts" in the February 2013 issue. I love the article! Mr. Henry has done a thorough job coming up with such a wide array of components that can be connected to the PIC. With regards to the serial port, a series protection resistor has been included. I do not understand how this would protect the microcontroller from the possible +12 volt and -12 volt signals. The resistor will limit the current, but wouldn't the microcontroller pin still be exposed to voltage that far surpasses the absolute maximum ratings, risking damage to the MCU?

Judy May W1ORO
Union, KY

A quick check of the PIC datasheet shows that most of the pins feature input protection diodes which will effectively clamp the voltages. The series resistor limits the current through them. I've never had any trouble with this scheme when sensing RS-232 signals.

Thomas Henry

Programming the Brains Behind CNC Milling Tools and Motors

Go to www.nutsvolts.com/index.php?/magazine/article/october2013_DesignCycle for any additional files and/or downloads associated with this article. You can also post comments.

What does your cell phone have in common with a CNC machine? Your cell phone enclosure was manufactured in a mold that was crafted using a CNC machine. Taking that thought a bit further, a specialized CNC machine drilled the holes in your cell phone's circuit board while yet another specialized CNC machine stuffed the phone's printed circuit board (PCB) with electronic components. If you have an inkling that we will be talking about CNC machines this month, you're close but no cigar. We are going to focus on programming the silicon brains behind the milling tools and motors.

CNC 101

CNC — or Computer Numeric Control — is applied to milling in the form of a computing device that is capable of converting machine tool language into physical motion. The complexity of the CNC computing device is directly related to the CNC mill's desired function.

A simple CNC mill can be driven by a microcontroller. More complex CNC mills are normally driven by PC-class systems running specialized CNC software.

PCs are wonderful application execution platforms. However, the performance of a standard off-the-shelf computer is horrible when it comes to high speed I/O operations required by CNC applications. To offset the PC's lack of I/O speed, a typical computer-based CNC system will include a drive which sits between the PC's limited I/O subsystem and the mill motor. Drives can be endowed with intelligence or simply act as dumb high-current motor controllers. In that a PC is not designed to supply pulse trains that roam in sub-millisecond space, an intelligent drive is most often used in

computer-based CNC systems. A popular intelligent motor drive is squirming around in **Photo 1**.

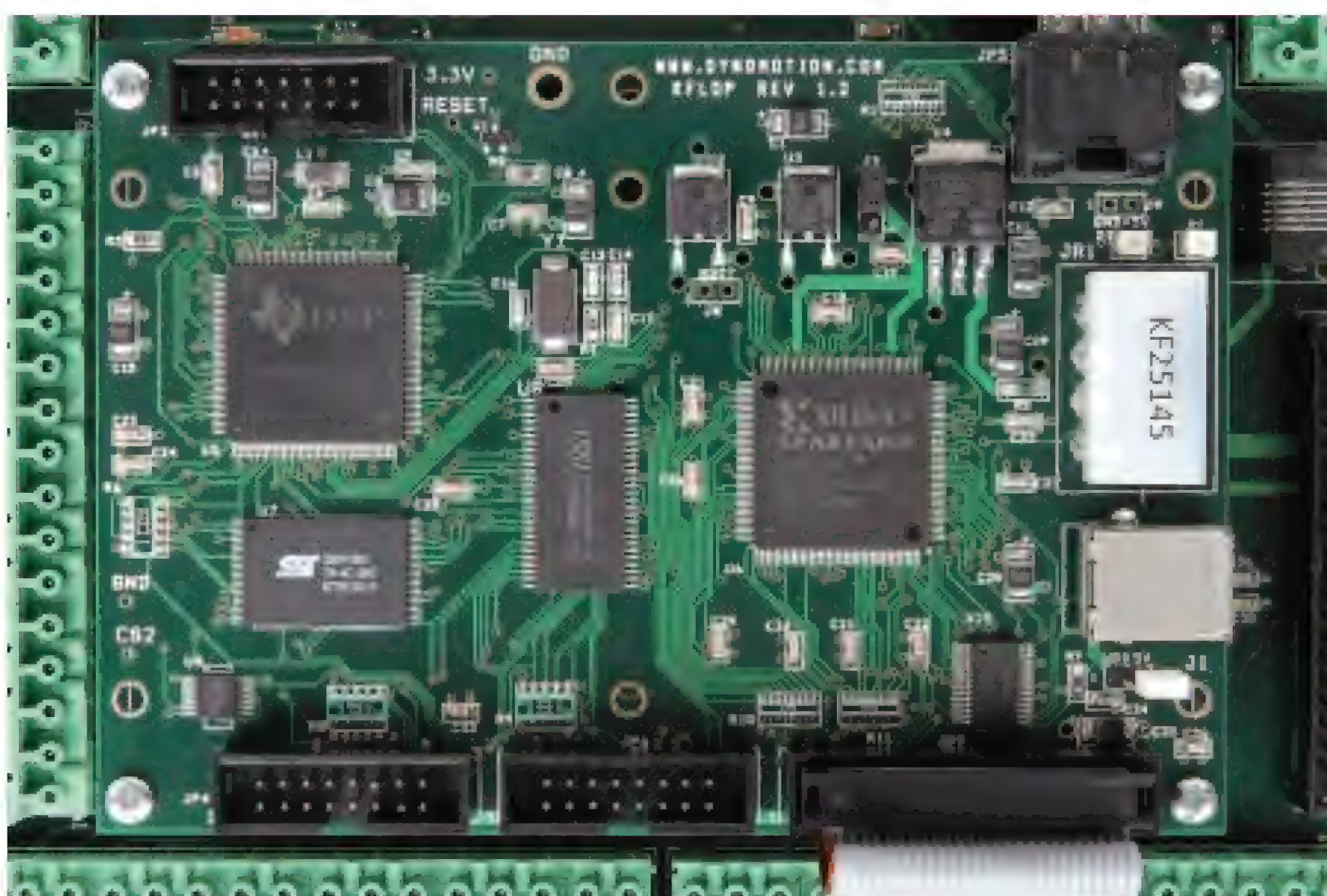
We are not going to get into CNC motors. However, the most commonly used motors in the CNC environment are servo motors and stepper motors.

CNC 201

Since the GECKODRIVE you see in **Photo 1** is dedicated to driving stepper motors, there isn't much more movement information we can provide to the stepper drive. All we are able to configure is the upper amperage limit by placing the



■ Photo 1. The GECKODRIVE you see here is designed to aid in the control of a standard stepper motor.



■ Photo 2. The DynoMotion KFLOP is a high powered embedded computing device that is capable of running multiple C subprograms (threads). The threads directly affect the behavior of the DynoMotion KFLOP's I/O subsystem.

appropriate resistor across the *CURRENT SET* pins.

If our PC happens to be old enough to have a parallel port, we could use the parallel port output pins to drive the GECKODRIVE's opto-isolated inputs. However, doing so would require a hefty amount of programming at the computer level. If we can't program what we need, we can throw money at the problem and purchase a fancy PC CNC application.

Today's PCs have dumped both the standard RS-232 port and the parallel printer port in favor of USB. USB allows us to drive external devices at very high data rates. However, do you see a USB portal on the GECKODRIVE? Although USB is a robust data channel, applying USB to a CNC application raises another possible computer-related problem.

To perform efficiently as a CNC controller, the PC must be equipped with enough resources (RAM, disk, CPU, etc.) to quickly make the necessary calculations



■ Screenshot 1. This view allows us to write our C program, compile it, and load it into temporary storage for execution.

required by the CNC application. Purchasing a high-end super fast PC is not the complete answer because we would still have to provide an interface between the PC USB portal and the GECKODRIVE's opto-isolated inputs. Since we're being forced to include an interface board, let's make it worth our while.

The DynoMotion KFLOP

The DynoMotion KFLOP shown in **Photo 2** augments a PC that is acting as a CNC controller. KFLOP also takes advantage of today's available PC hardware. The PC CNC controller communicates with the KFLOP via USB. KFLOP USB duties are handled by an FTDI FT245R USB FIFO IC. The remaining KFLOP electronics include a Texas Instruments TMS320C6722 DSP, a Xilinx FPGA, a Microchip 1M x 16 Flash memory IC, and an ISSI 128M x 16 SDRAM.

The DynoMotion KFLOP can perform 32-bit and 64-bit floating point math without the assistance of the attached PC. Everything a CNC motor needs in life is part of the KFLOP's programming. The DynoMotion KFLOP even comes with its very own C compiler. The USB connection is simply a link to the KFLOP's monitor and control program that is running on the PC.

As you can see in **Photo 2**, there isn't very much KFLOP hardware to talk about. Don't let that mislead you. Connector JP7 (lower right of **Photo 2**) houses 18 LVTTTL I/O pins. There are 10 LVTTTL I/O positions at JP4 (Aux #0) and 10 more at JP6 (Aux #1). Nine more LVTTTL I/O positions are housed in the female RJ-45 connector JP5 (upper right of **Photo 2**). JP2 is the JTAG interface. We won't be utilizing the services of JTAG here.

Now that you are aware of the DynoMotion KFLOP's resource-rich I/O subsystem, let's put it through its paces. All we need as a prerequisite for manipulating the KFLOP I/O subsystem is a basic knowledge of C programming.

There are a bunch of I/O positions to keep up with. Not to worry. The KMOTION IDE (Integrated Development Environment) allows us to keep tabs on every bit of the KFLOP I/O. KMOTION provides access to the KFLOP C coding and hardware programming environments. It is also possible to do some G code

processing from within KMOTION. We can even adorn our CNC C programs with human readable messages. **Screenshot 1** is a capture of the C program function of the KMOTION IDE. The *KMotionDef.h* Include file contains all of the constants, macros, and data structures we can include in our C programs to influence the KFLOP I/O pins. For you C programmers out there, this is ground floor programming. The comment suggests that the message will be displayed on a console. **Screenshot 2** contains the result of creating, compiling, downloading, and running our little *DesignCyclePrint.c* code.

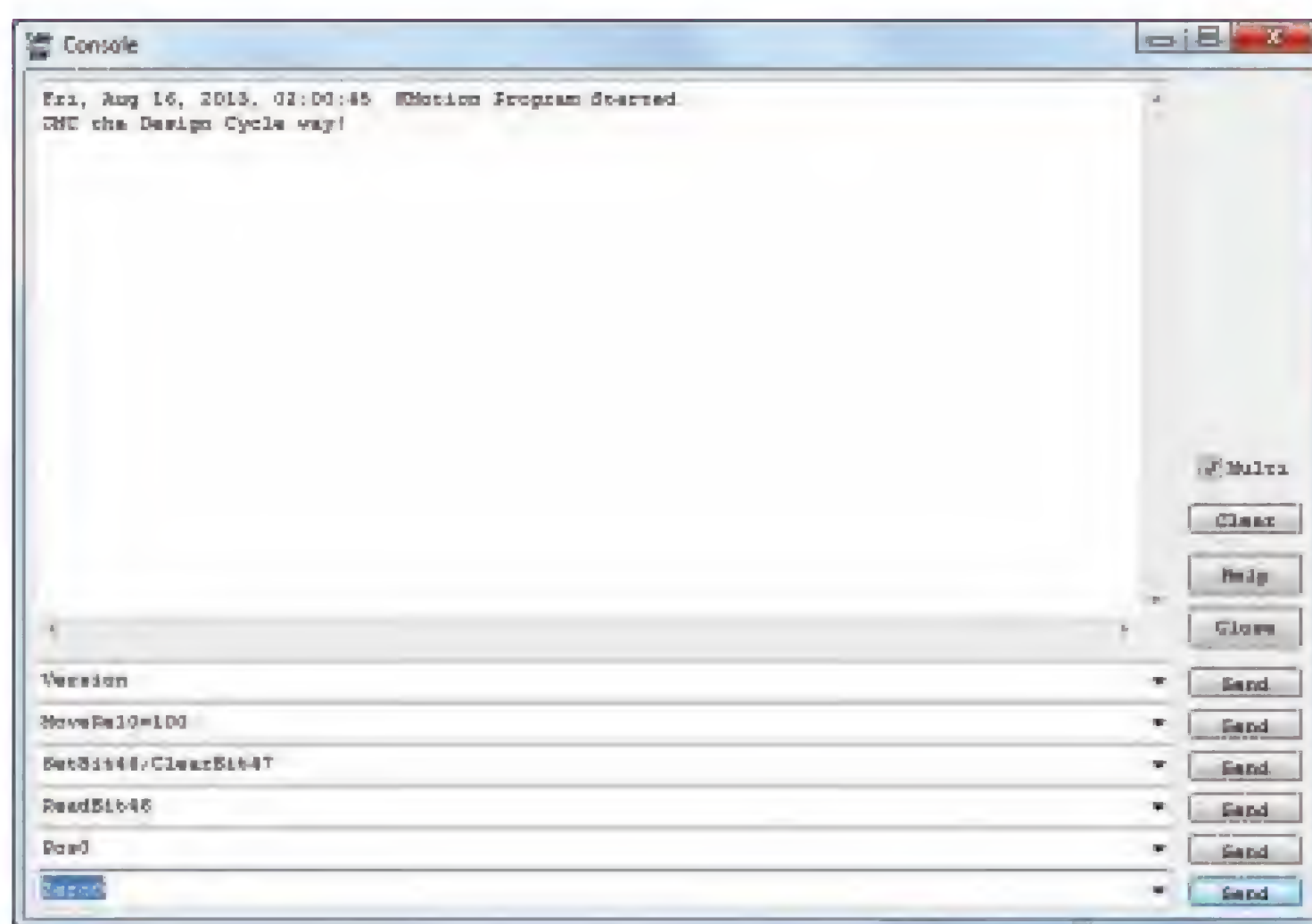
Take another look at **Screenshot 1** and note that the collection of arrows at the far right of the C program view's toolbar performs the compile, download, and run steps with one mouse click. The downloaded object code is volatile and will not persist through a KFLOP power-on or reset operation. However, we can control the execution of the downloaded code with the Run and Halt commands.

If we decide to store our *DesignCyclePrint.c* object code in program Flash and launch the code thread at power-up, we would simply fill the Thread 1 check box and click the User Memory button in the Flash panel of **Screenshot 3**. Take yet another look at **Screenshot 1** and you will see that our *DesignCyclePrint.c* code is contained in Thread 1. You will also notice that we can load and execute up to seven threads. The KFLOP employs time slicing to allow for all of the threads to execute.

The Channel dropdown window captured in **Screenshot 3** allows us to configure up to eight channels. A KFLOP channel is actually an axis (X, Y, Z, etc.). Thus, we can configure up to eight axes, which is the KFLOP hardware limit. The Axis Mode input and output selections are assigned to physical I/O pins with the Input Channels and Output Channels panels. In the case of the Input Channels and Output Channels, the channels are not axes. Input and Output Channels are directly related to KFLOP I/O pins.

There are 16 possible Input Channels and 32 possible Output Channels per axis at our disposal. The relationship between axes, axis modes, and I/O channels becomes evident when you start assembling the physical encoders, drives, and motors that will make up your CNC system. As you select options contained within the Configuration view, the KMOTION application is simultaneously generating related C source code and placing it on the Clipboard. The format of the KFLOP-generated code is described in the *KMotionDef.h* include file. By simply clicking the C Code -> Clipboard button, the resultant C source code stored on the Clipboard can be moved into a thread view for editing, compilation, and loading.

Before committing your code to Flash, you must first compile and load the release version of your code into the KFLOP. Clicking on the *User Memory* button programs,

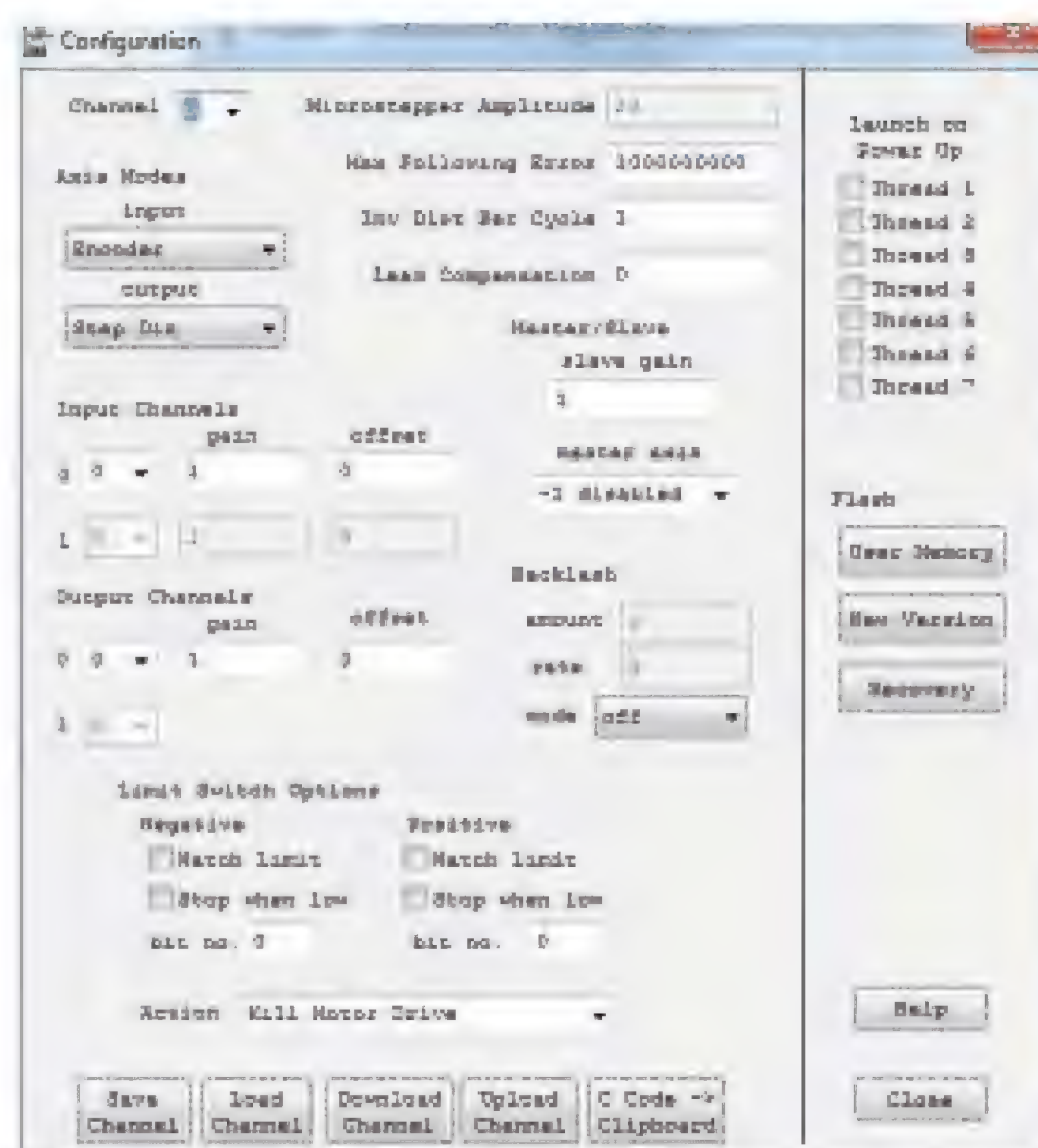


■ Screenshot 2. The console provides instant C programming gratification. The console view can be used to debug our C code or keep us informed as to the status of a running KFLOP application.

the downloaded threads into the KFLOP's program Flash. Any or all of the seven available threads can be specified to load and execute at power-up. Threads can also be started and stopped under program control.

KFLOP Base I/O

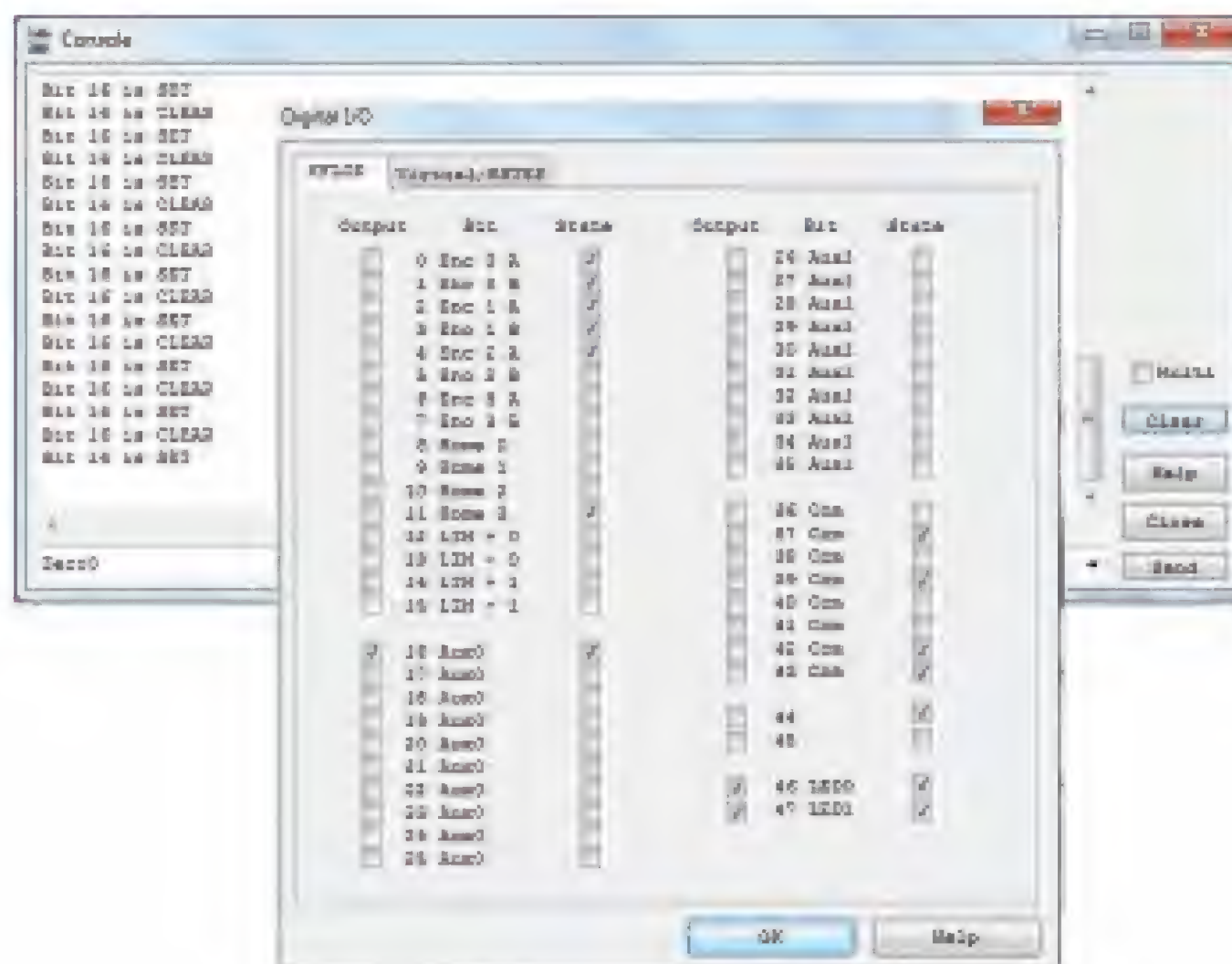
As you can see in **Screenshot 4**, the KFLOP is specifically designed to be used in a CNC environment.



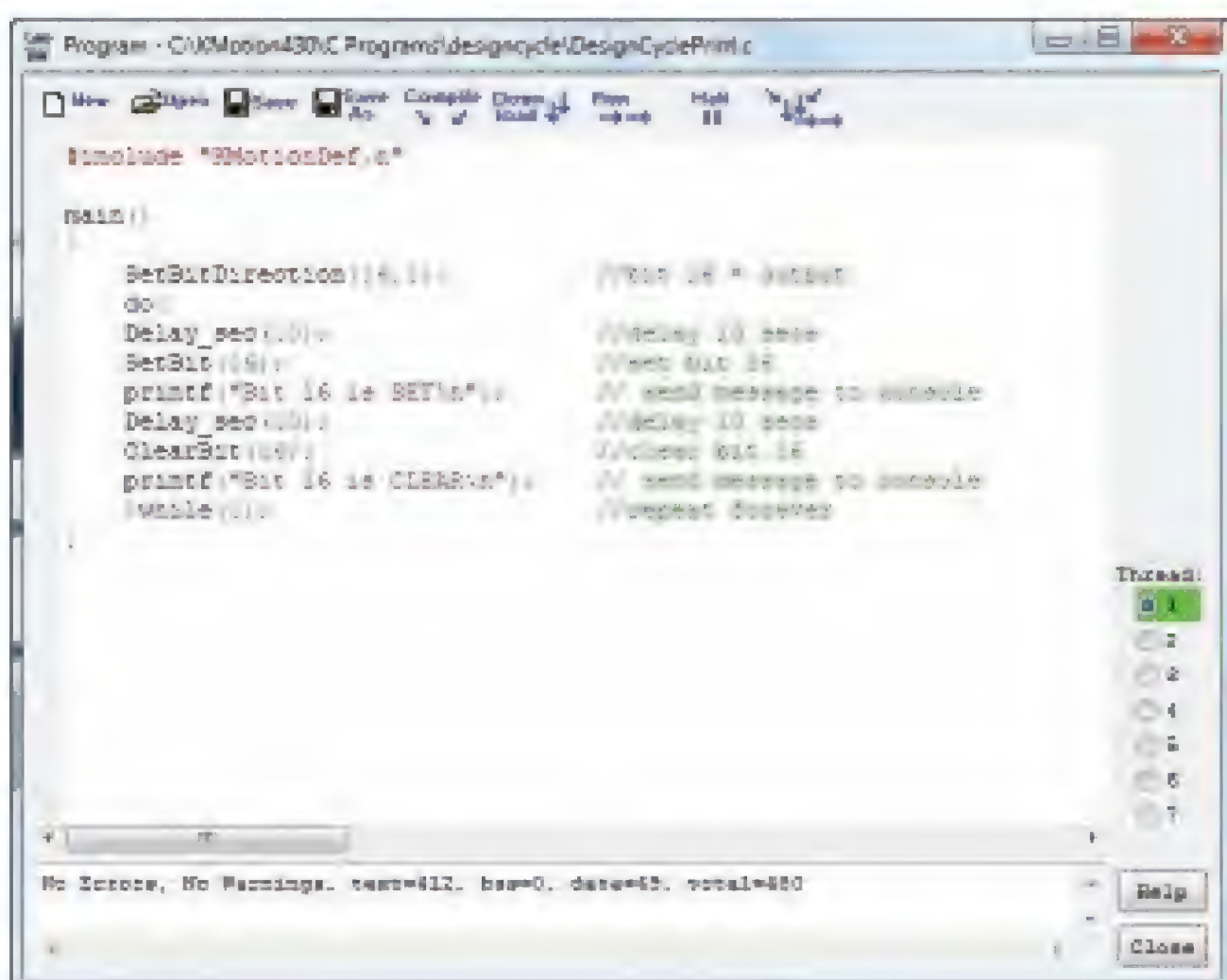
■ Screenshot 3. This view will mean more when you get to the point of actually building the CNC hardware system. For now, all we care about is the Flash panel which allows us to load a new version of KFLOP firmware and load our C object code into program Flash memory.



■ Screenshot 4. The KFL0P I/O pins listed here can be configured as outputs by simply clicking the Output check box associated with the I/O pin.

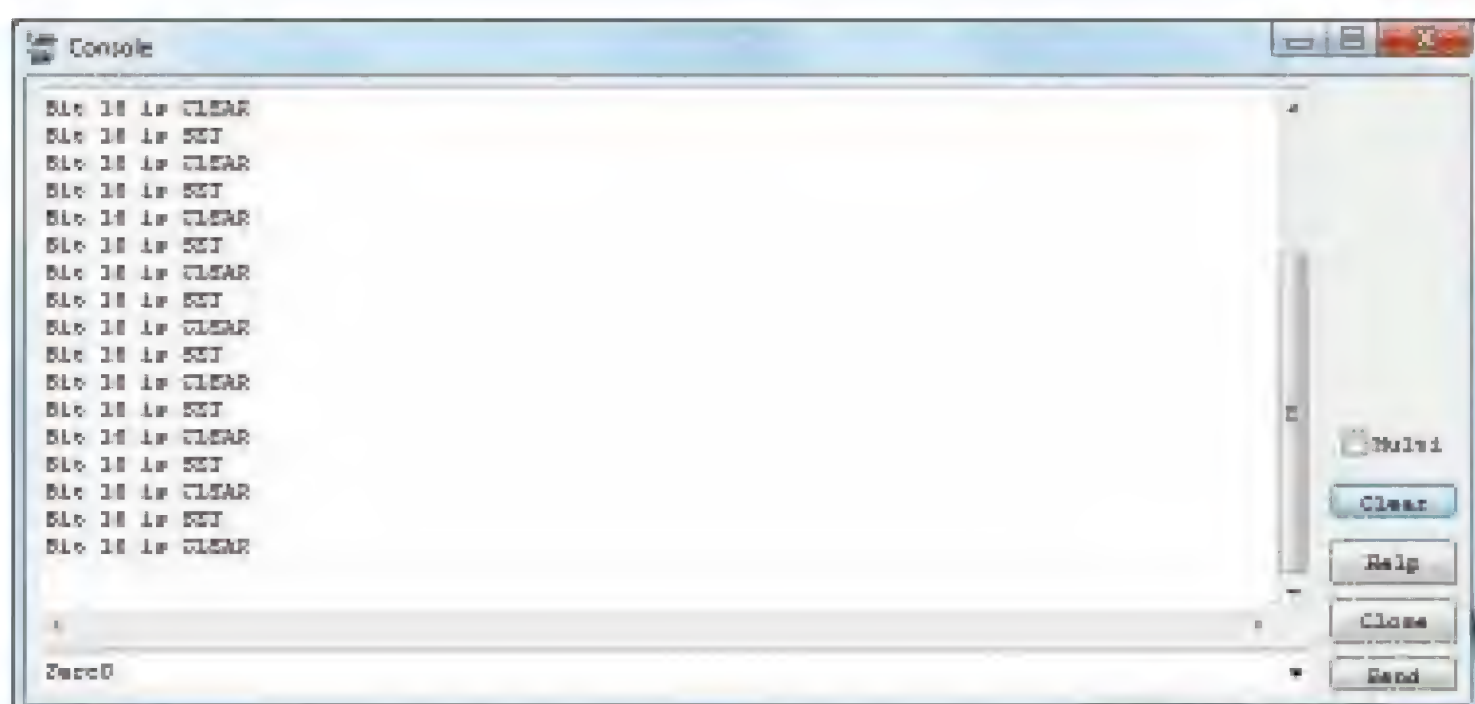


■ Screenshot 7. This view of the KMOTION console and digital I/O panels is displayed when the C program sets I/O bit 16.



■ Screenshot 5. The example C program shows just how easy it is to manipulate the KFL0P's I/O subsystem.

The first 16 I/O positions are dedicated to encoders, home switches, and limit switches. I/O bits 46 and 47 are dedicated to a pair of user-programmable LEDs. You can see the LEDs that are placed just above the



■ Screenshot 6. So it is written. So shall it be displayed.

power connector (JR1) in **Photo 2**.

The I/O pins that make up JP4 (Aux #0) are shown in **Screenshot 4** as KFL0P I/O pins 16 through 25. The KFL0P's JP5 I/O pins are listed as 26 through 35. KFL0P I/O pins 36 through 43 are part of the LV differential connector JP5. The JP5 I/O pins are intended to be used as high speed differential communications pins that interface using twisted pair cable. I/O bits 44 and 45 are located in the JP7 I/O pin cluster. All of the JP7 I/O pins are five volt tolerant with the exception of I/O pins 44 and 45 which are restricted to 3.3 volt logic levels.

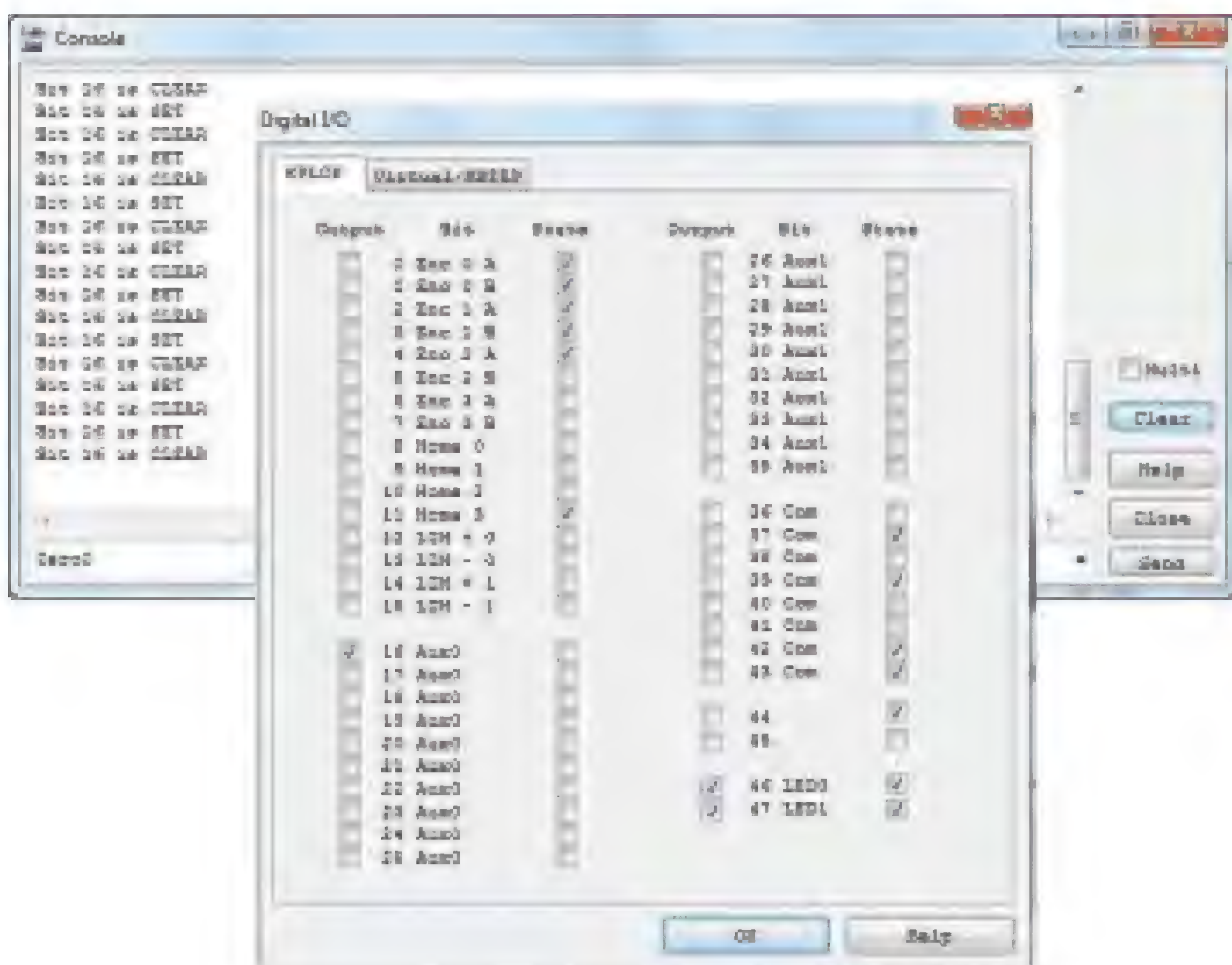
KFL0P I/O pins can be controlled directly from the digital I/O view or programmatically using a C program or script command. For instance, I/O bit 16 can be forced into output mode by simply clicking the bit 16 Output check box.

The logical state of bit 16 is displayed in the bit 16 State check box. We can also control the direction and logical state of I/O bit 16 from within a C program thread. Take a look at **Screenshot 5**. Thread 1 now contains a C program that will manipulate I/O bit 16. The program is pretty easy to follow.

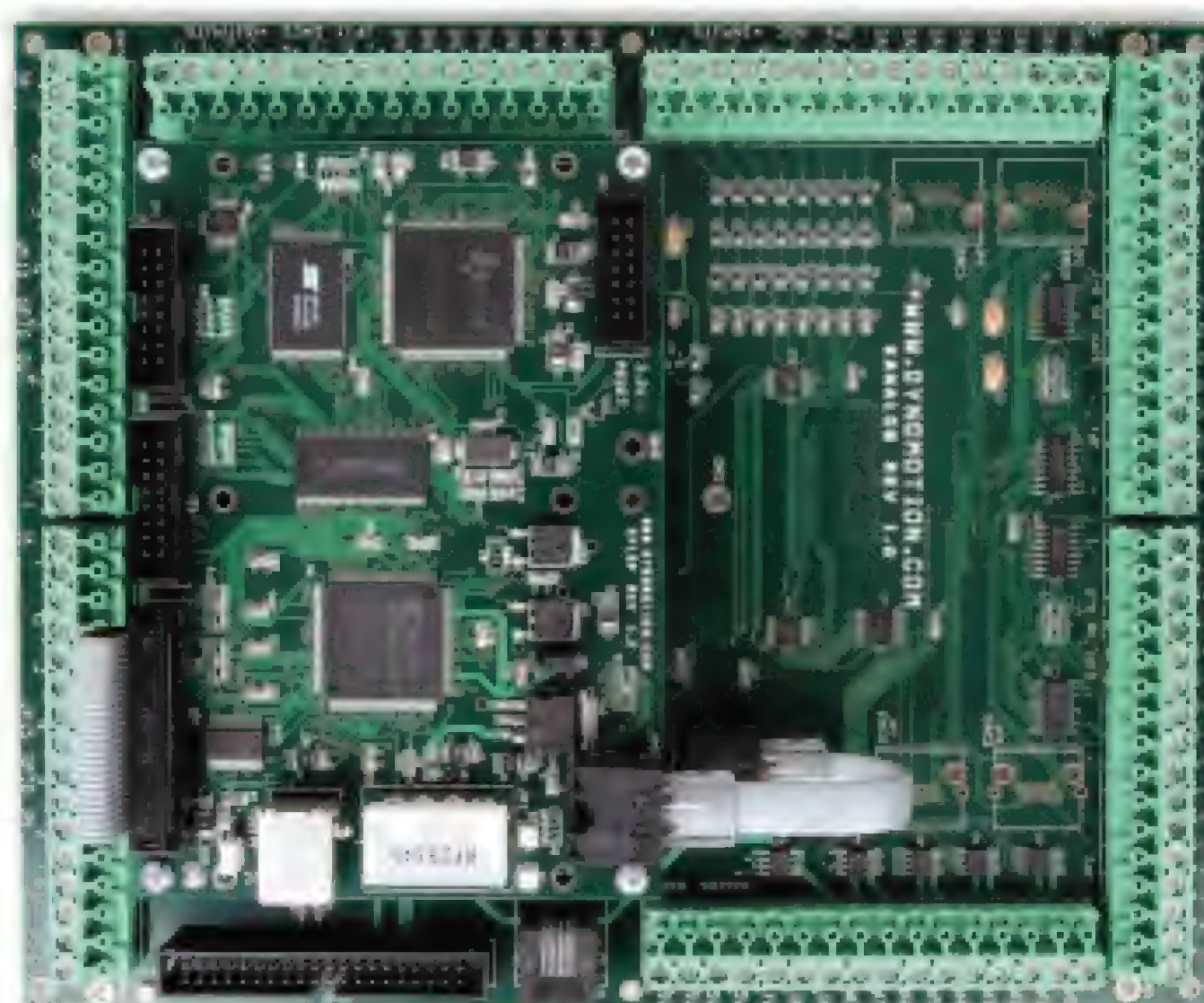
We set I/O bit 16's direction to output and alternately toggle the bit's output state every 10 seconds. Our C program also provides a human readable output describing the current state of I/O bit 16.

The console view associated with our bit toggle program is shown in **Screenshot 6**. Note that the bit 16 Output check box is checked in **Screenshots 7** and **8**. Bit 16 is forced logically high in **Screenshot 7** and taken logically low in **Screenshot 8**.

As you can see, the State check box corresponds to the information in the Console view.



■ Screenshot 8. This is what the user/programmer sees when the C program clears KFLOP I/O bit 16.



■ Photo 3. Kanalog combines the compute power of the KFLOP with optoisolated inputs and outputs, differential encoder inputs, analog-to-digital inputs, FET relay drivers, and digital-to-analog outputs.

KFLOP Does Analog Too

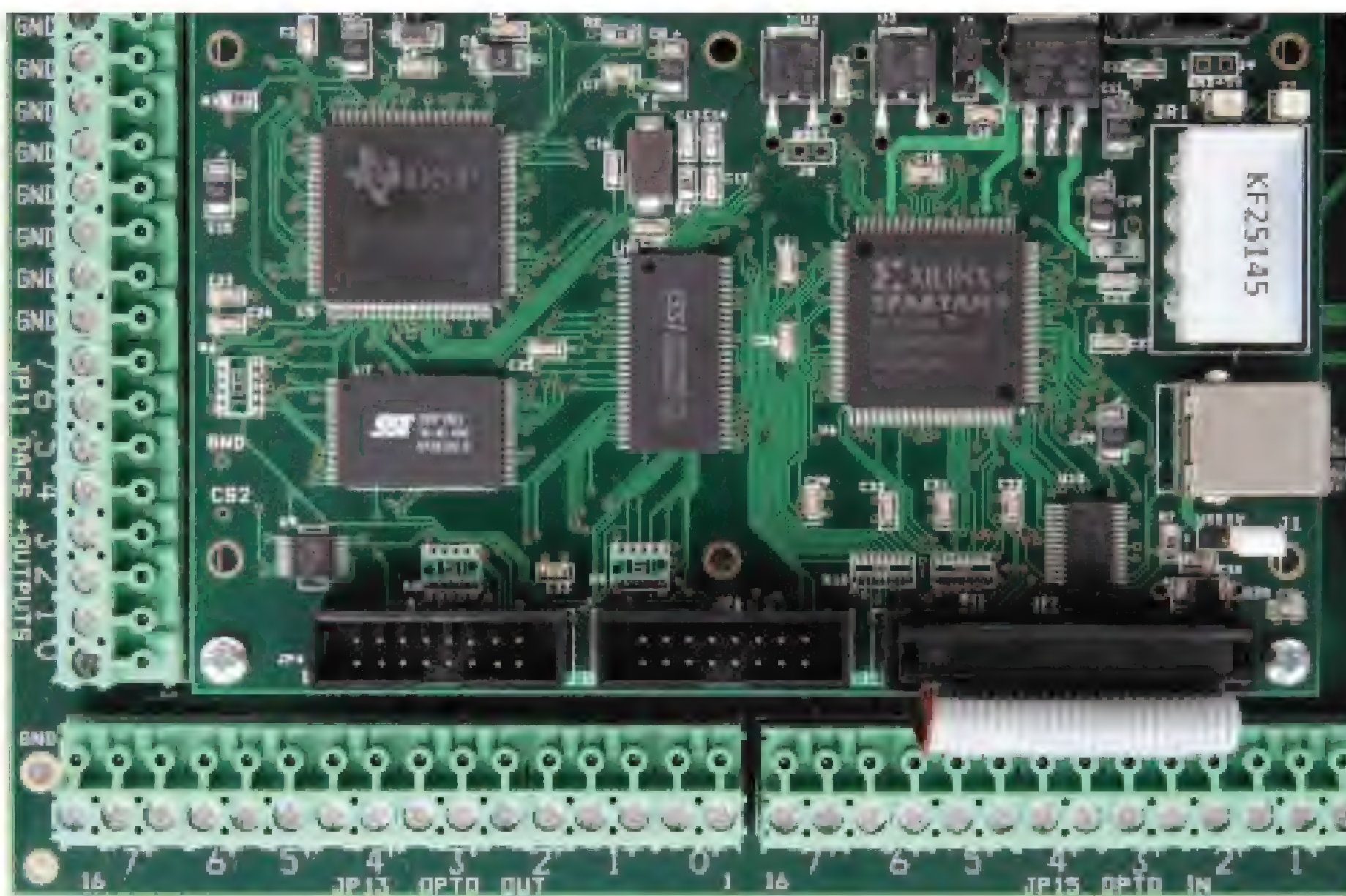
With a little help from its friends, flying a bit higher over **Photo 2** reveals a larger I/O interface board underneath the KFLOP. The larger board that is carrying the KFLOP in **Photo 3** is called Kanalog. The Kanalog attaches to the KFLOP using the high speed LVTTTL differential communications portal JP5 and general-purpose I/O portal JP7. It may seem that we have lost some KFLOP I/O pins to the Kanalog. However, Kanalog is actually an I/O pin force multiplier. Let's take a walk around the Kanalog, beginning with the optoisolator I/O interface you see in **Photo 4**.

Kanalog provides eight opto inputs and eight opto outputs. There is nothing special about the opto configurations. Kanalog uses optoisolators in the standard manner. The opto inputs are no more than LEDs with a 10K series resistor. The opto inputs can be driven with voltages that range from +5 to +24 volts.

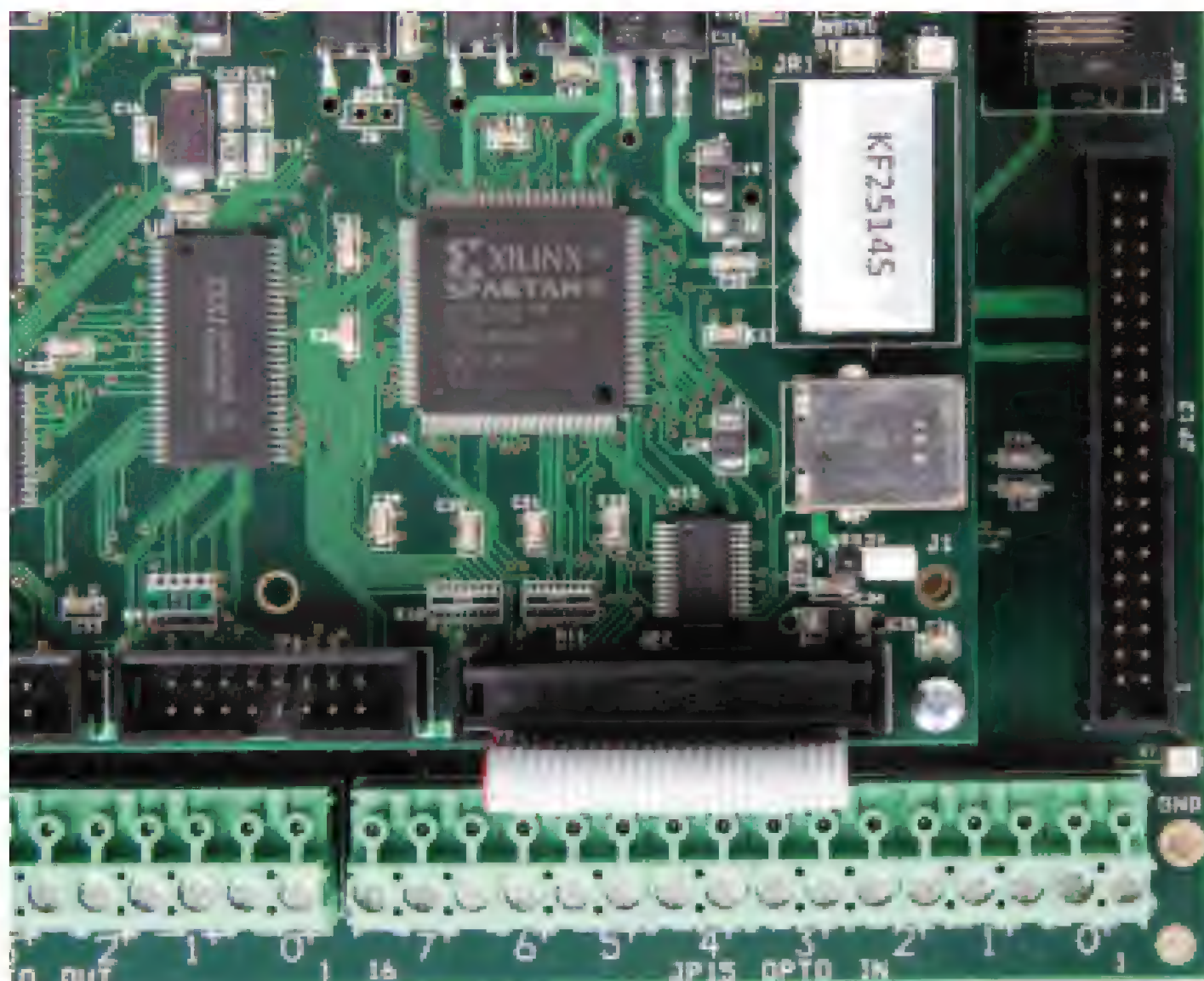
Note the opto inputs are pinned out as eight pairs of plus/minus inputs. The opto outputs are also pinned as eight pairs of plus/minus outputs. The JP13 and JP15 opto plus/minus arrangements pin out the LED anodes and cathodes, respectively. NPN transistors make up the opto outputs with the plus pins attaching to open collectors, and the minus pins

connecting each respective optoisolator transistor emitter. Each opto output is rated to handle up to 80 volts at 25 mA.

JP12 is a standard 40-pin IDC header that houses eight bits of 3.3 volt LVTTTL input and eight bits of 3.3 volt LVTTTL output. There are also four 0.0-3.0 volt analog-to-digital converter (ADC) inputs among the 40 pins. The Kanalog has an onboard ± 15 volt @ 70 mA DC-DC



■ Photo 4. Note the ribbon cable interface to the Kanalog mounted in JP7. Kanalog provides eight opto inputs and eight opto outputs. You can also see the DAC outputs in this shot.



■ Photo 5. JP12 is the Kanalog non-isolated GPIO interface point. We can also access the 3.3 volt and 5.0 volt power rails via JP12.

generator whose output pins are also part of JP12. The gate and drain of an extra 24 volts @ 1A FET tops off the

JP12 pinout. A visual of JP12 is provided in **Photo 5**.

Photo 6 covers a lot of territory. The 16 differential encoder inputs are located across the very top of the shot. The first eight differential encoder inputs communicate with the KFLOP via a short ribbon cable connected to the Kanalog JP14 connector. The second set of eight differential encoder inputs communicate by way of the RJ-45 terminated silver satin cable.

The FET relay drivers are accessed via JP8 at the far right. Each FET is rated for 24 volts @ one amp. The FETs are wired to provide a ground at the SWx pins when the associated I/O bit is activated. There's no rocket science here. A positive power supply is connected to one side of the relay coil and the other side of the relay coil attaches to an SWx pin. Kanalog does not include a reverse diode for the relay.

So, we must be sure to install the reverse diode across the relay coil to prevent damage to the Kanalog circuitry. If the FET switch is not driving an inductive load, the reverse diode is not necessary.

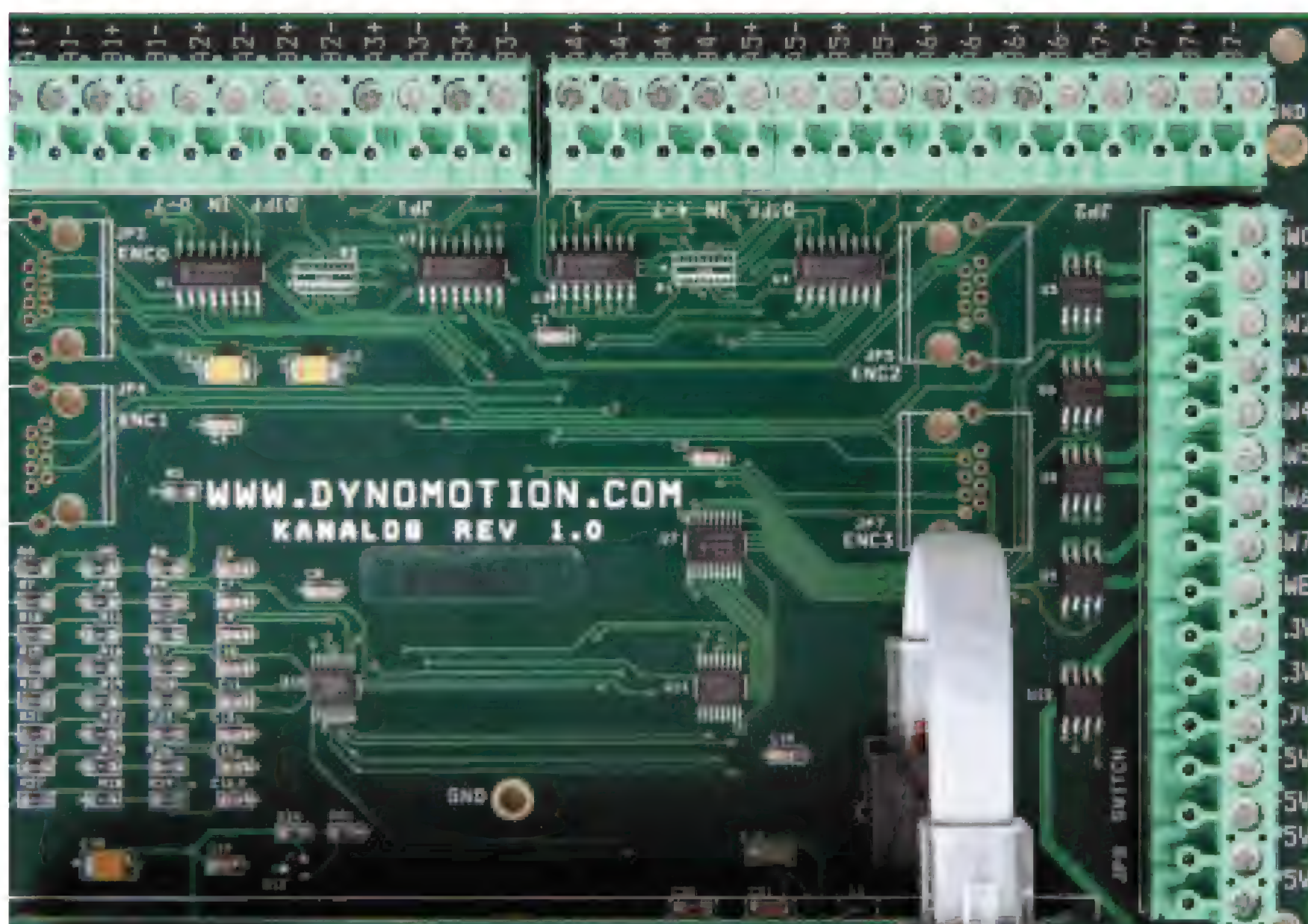
The Kanalog ADC inputs and digital-to-analog converter (DAC) outputs are rated for ± 10 volts. The ± 10 volt levels are common to CNC control and monitoring peripherals. **Photo 7** lays out the ADC

interface points. You can look back and take a look at the DAC outputs in **Photo 4**.

To bring up the Kanalog in KMOTION, we must activate the Kanalog in the KMOTION Options drop-down menu. Once that's done, a Kanalog tab is added to the digital I/O view and the Kanalog ADC and DAC values become available in the analog I/O view.

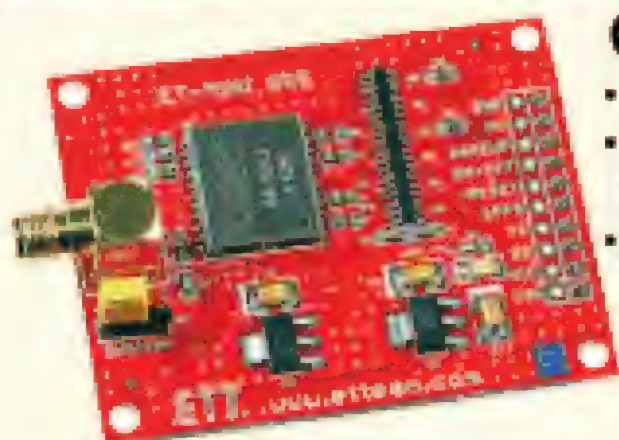
Screenshot 9 visually lays all of this out. All of the I/O bits in the left-most column of the digital I/O view are inputs, while all of the Kanalog outputs are listed in the right column.

We can easily put a finger on the Kanalog's opto I/O, ADC, DAC, and FET driver interface points. However, the general-purpose bits are not that obvious. Input bits 128 through 135 and output bits 160 through 167 are positioned on the JP12 connector. JP12 pin mapping can be found in the KMOTION manual package which can be downloaded from the DynoMotion website.



■ Photo 6. The 16 differential encoder inputs and FET relay drivers are captured in this shot. The RJ-45 connection enables the second set of eight differential encoder inputs. The FET relay interface is positioned at the far right.

Save Up To 60% On Electronic Components



GPS Mini Board

- Ideal for Adding GPS to your System
- Connects Directly to Most Microcontrollers and Arduino Boards
- Position Accuracy to 2.5m

Only \$64.90

New 10A Solar Regulator

- High Performance Microcontroller Based PWM Regulator
- Suitable for both 12V and 24V Systems
- Works with Gel, Sealed and Standard Lead-Acid Batteries

Only \$19.90



PIC18F4550 Controller

- Powerful New Controller Board Based on the Microchip PIC18F4550
- Heaps of I/O including a USB Connection
- A/D, EEPROM, LED's and In-Circuit Programming

Only \$29.90

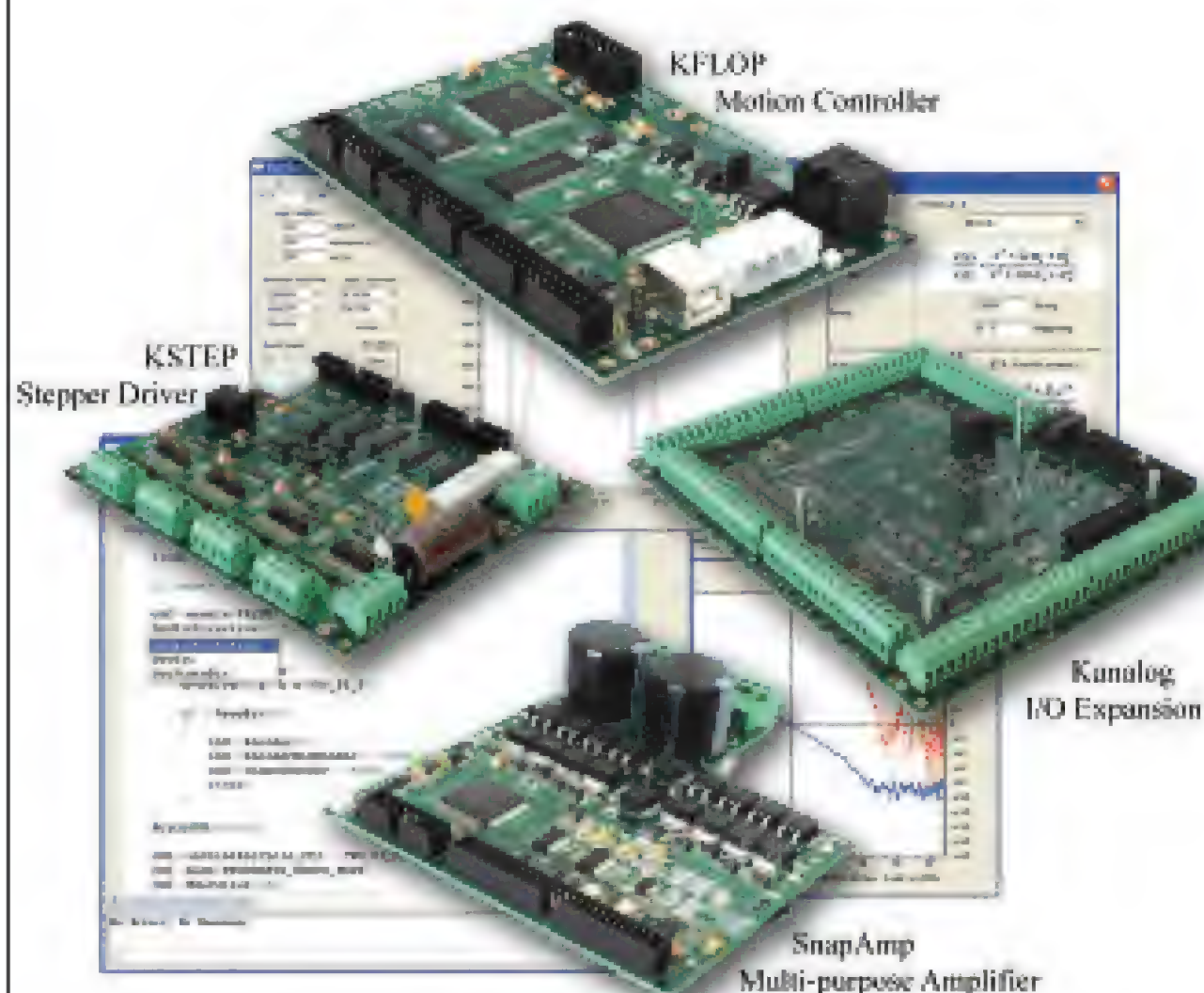


We are your one-stop shop for Microcontroller Boards, PCB Manufacture and Electronic Components

www.futurlec.com

DYNAMOTION

Premium Motion Control for CNC Manufacturing, Robotics and Automation



With a complete ecosystem of boards and software DynoMotion has something for everyone. Visit our website to learn more.

www.Dynomotion.com



■ Photo 7. This shot reveals the remaining differential encoder inputs and the ADC converter input points.

Instant Gratification

The KMOTION-KFLOP-Kanalog combination allows us to quickly whip up some C code and exercise the functionality of the KFLOP and Kanalog. For instance, let's see what it takes to force the Kanalog DAC to output

+10 volts. Hide the women and children as the C program is frightening. Check out **Screenshot 10**. The C program instructs the channel 0 DAC to load a value of 2048, which results in +10 volts out. The DAC mnemonic is actually a function name. The DAC function and its argument limits are outlined in the *KMotionDef.h* file. I simply slung a wire between DAC output 0 and ADC input 0 to get the ADC reading.

Don't get hung up on the Kanalog's DAC and ADC accuracy. Remember that in the CNC world, the encoders and motors are the precision instruments. DACs and ADCs are not normally directly involved in the precision machining process.

CNC Specifics

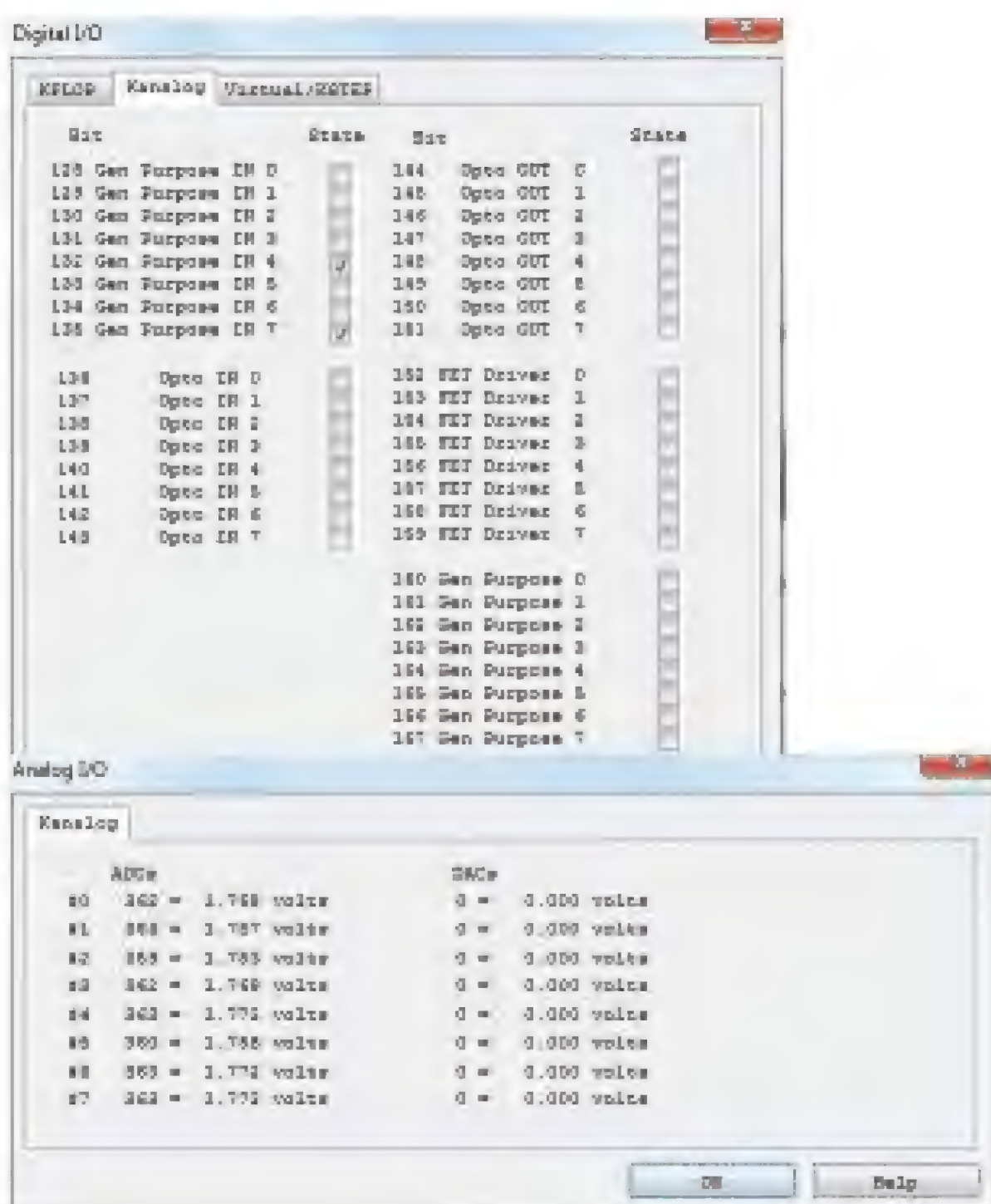
The KFLOP uses the services of a plugin to pass data elements to and from third party applications running on the PC.

DynoMotion offers a PC CNC program that is designed to complement the KFLOP and Kanalog. Mach 3 is another popular third party CNC application that has proven to be KFLOP-friendly.

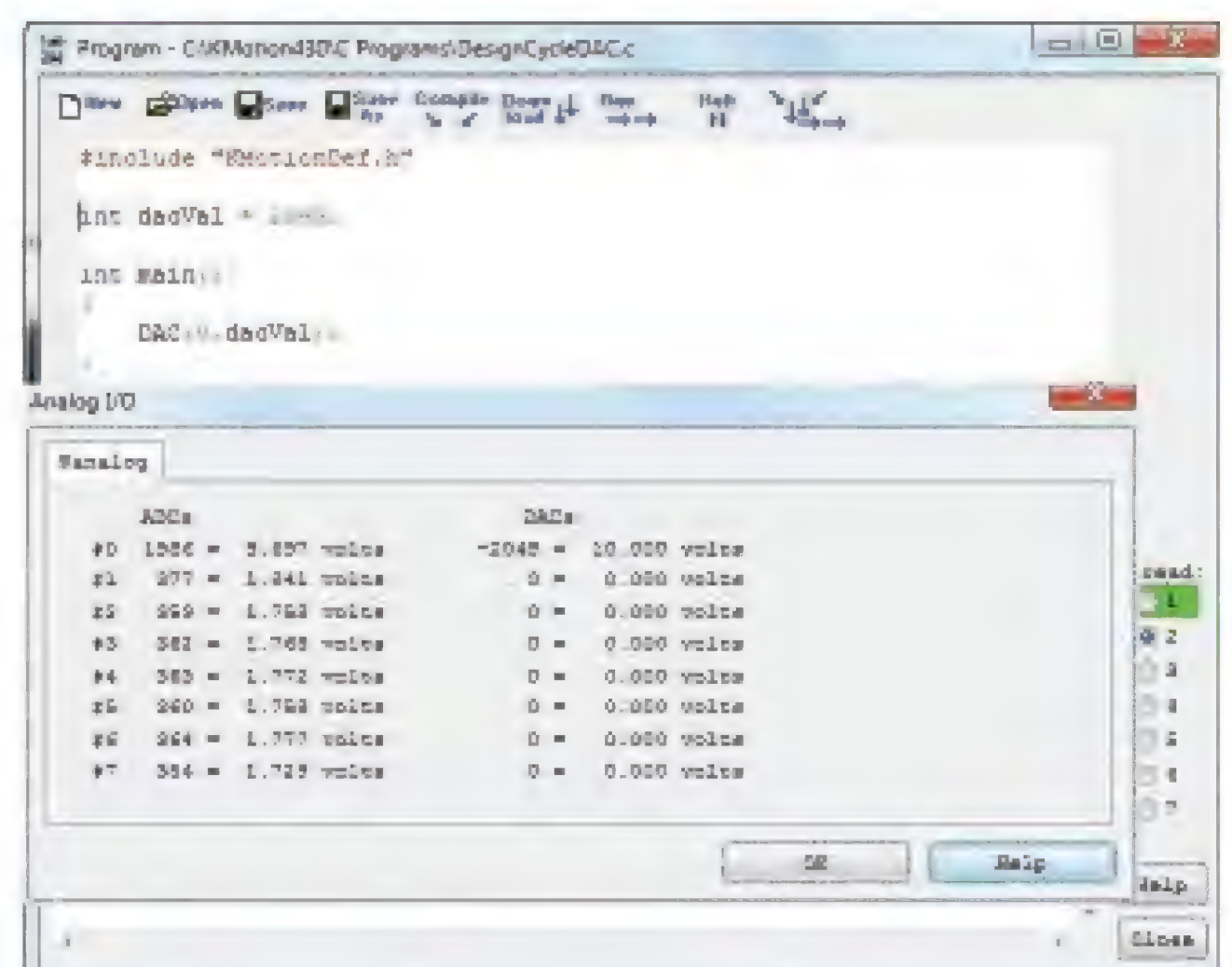
Time to Move On

Whether you are planning on using a KFLOP in a custom CNC application or applying the KFLOP to an embedded monitor and control project, you now have enough KFLOP/Kanalog knowledge to be successful. You can add the KFLOP and Kanalog to your design cycle.

NV



■ Screenshot 9. The Kanalog I/O pins and analog subsystem cannot be manually operated. We must use the bit operations supported by the C compiler to effectively use the Kanalog resources.



■ Screenshot 10. The Kanalog analog I/O view provides instant gratification for the ADC and DAC functions.

Arduino Handheld Prototyper

Part 4 Fresh Air Controller

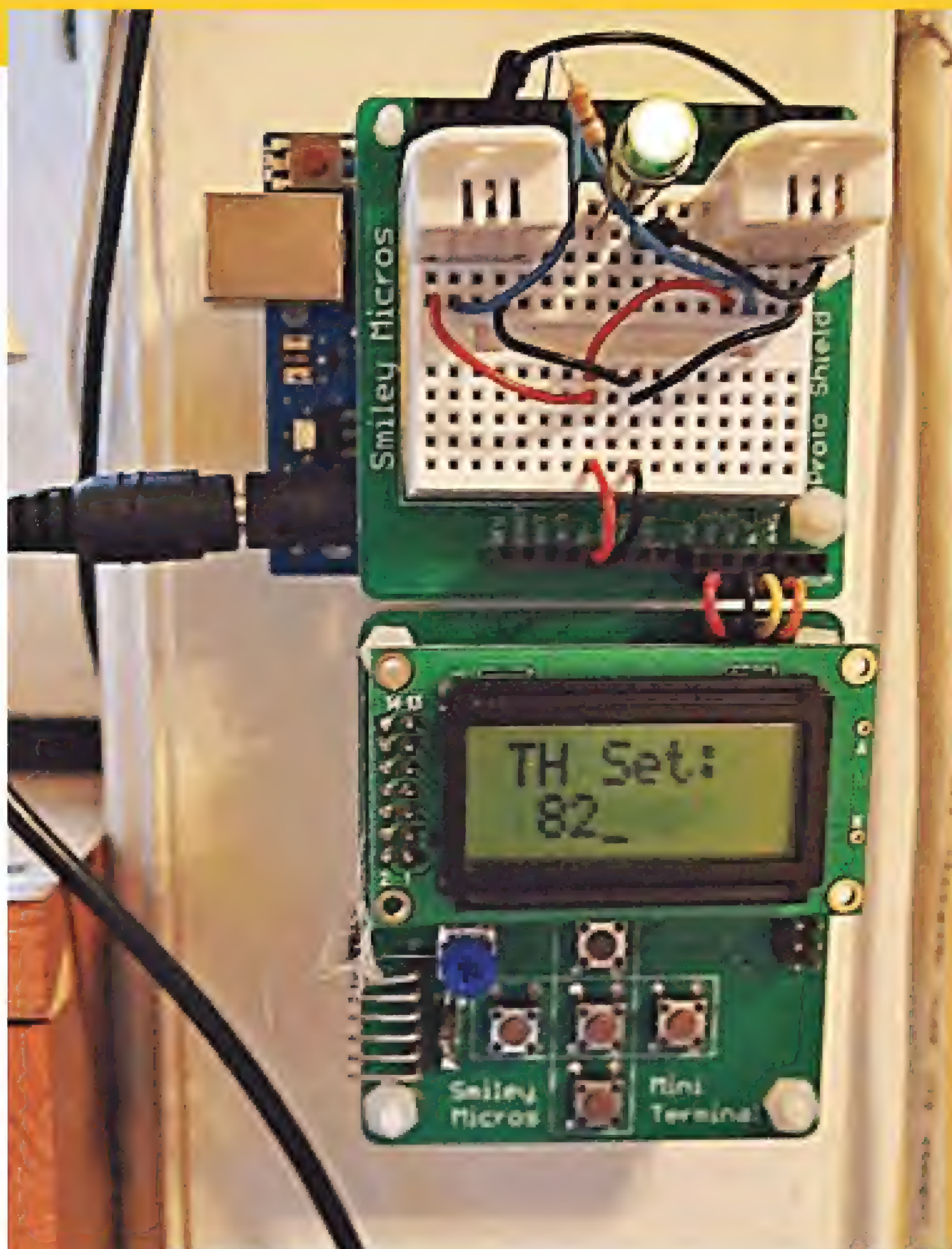
Over the past several months, we have been learning to use the Arduino handheld prototyper — a device that lets us design our prototypes on a breadboard with the Arduino proto shield and communicate with the I²C mini terminal — both of which are tied together with a plastic base so that you can carry the entire development system around in your hand (or hang it on a wall as shown in **Figure 1**). Last month, we began looking at a design using this system for a fresh air controller. This month, we will finish that design — mainly seeing how to communicate with the user via the I²C mini terminal LCD and pushbutton keys. Since we pretty much stopped last month in mid-sentence talking about the software, I recommend you reread that article. (If you don't have it, you can find it at www.nutsvolts.com.)

Talking to the User

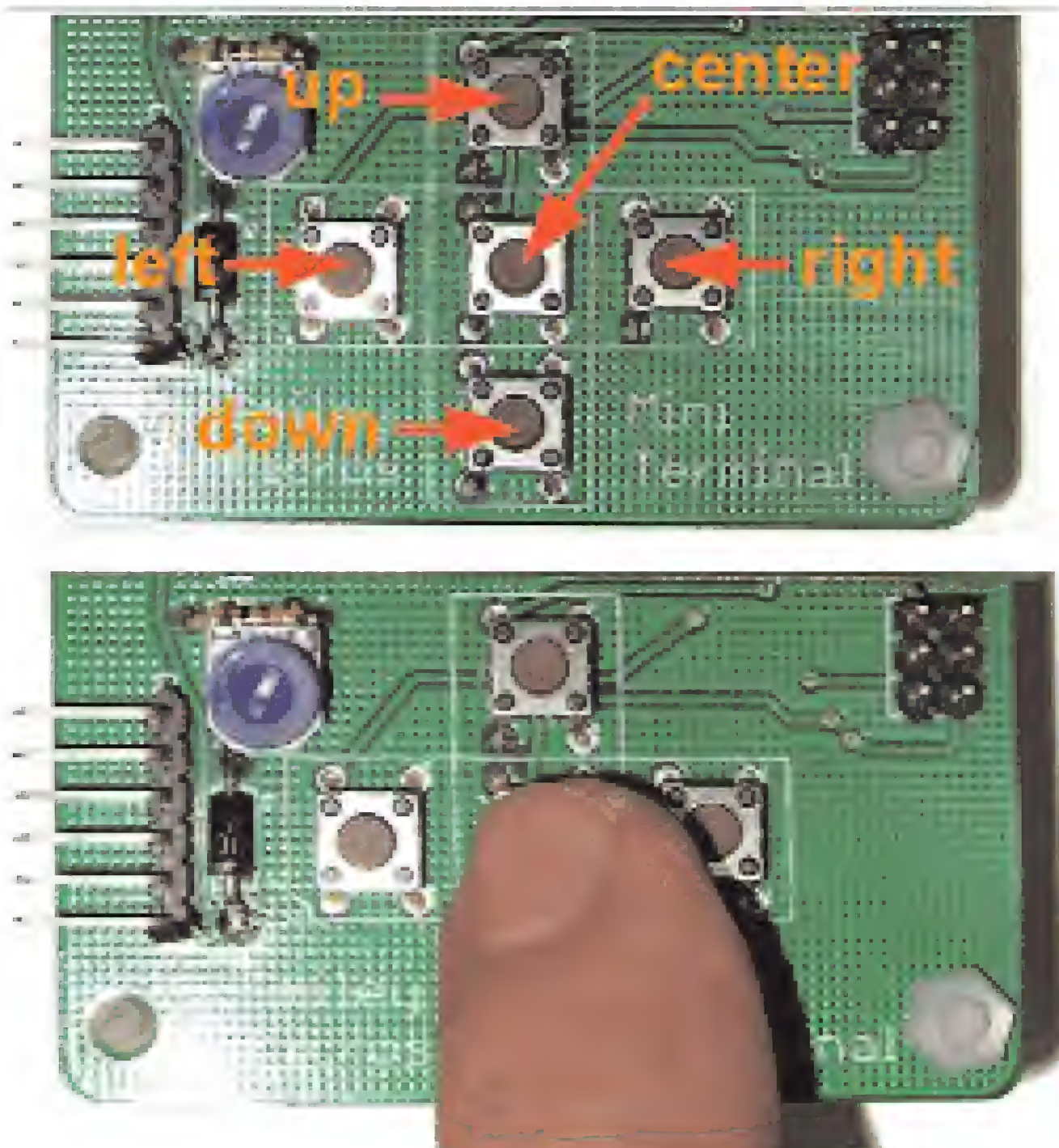
Last month when we discussed the fresh air algorithm, we specified that the user should set the variables for the high and low temperatures and the humidity. We could do this via the Arduino USB over the serial port of a PC but then it wouldn't really be 'handheld,' would it? For the system to be portable, we have to cut the apron strings from mommy PC and go it alone. To do this, we will use the I²C mini terminal part of the Arduino handheld prototyper.

We introduced how to use the I²C mini terminal over the past two articles; now we will see how to apply it to a real world application: Getting user feedback from the keys and displaying useful information on the LCD for the fresh air controller. We will write a program that has two basic modes: an idle mode for when the user is not accessing the system; and a menu mode for when the user wants to access the system by pressing a key.

The decision for which mode to run is made in the Arduino *loop()* function where we run the *idle()* routine until we receive a center (CTR) key press (**Figure 2**). Then, we run the *menu()* routine to get the user input. When the user is finished, we return to the idle mode. The *loop()* function is written as follows:



■ FIGURE 1: Fresh air controller prototype.



■ FIGURE 2: Press the center key.

```
void loop()
{
  if(idle()) // returns 1 if any new key is
             // available
  {
    if(newKey)
    {
      if(newKey == CTR)
      {
        menu(); // Lets us set temps and
                // humidity
      }
    }
  }
}
```

The Idle Mode

This system is going to be sitting around doing nothing most of the time. Oh, it checks the sensors every so often and turns the attic fan on and off occasionally, but mostly it just twiddles its digital thumbs. We can use some of that excess processing power by displaying a rotating series of LCD messages to show the user what the various indoor and outdoor temperature and humidities are, along with the various set points.

The *idle()* function does some stuff that — among other things — includes checking to see if a key is pressed. If the button is pressed, it returns 1; otherwise, it returns 0. The ‘some stuff’ mainly consists of showing the user some data on the LCD and checking the control algorithm to see if the fan should be on or off:

```
// Show the existing data and wait for a
// keypress
// void return if a key is pressed
#define IDLEDELAY 500
uint8_t idle()
{
  mt_printPM(tempOut,0,0);
  mt_print8BitNumber(tempOutdoor,0,1);
  if(checkKeyDelay(IDLEDELAY)) return(1);
  mt_printPM(tempIn,0,0);
  mt_print8BitNumber(tempIndoor,0,1);
  if(checkKeyDelay(IDLEDELAY)) return(1);
  mt_printPM(humOut,0,0);
  mt_print8BitNumber(humidityOutdoor,0,1);
  if(checkKeyDelay(IDLEDELAY)) return(1);
  mt_printPM(humIn,0,0);
  mt_print8BitNumber(humidityIndoor,0,1);
  if(checkKeyDelay(IDLEDELAY)) return(1);
  mt_printPM(tempHighSet,0,0);
  mt_print8BitNumber(setTempHigh,0,1);
  if(checkKeyDelay(IDLEDELAY)) return(1);
  mt_printPM(tempLowSet,0,0);
  mt_print8BitNumber(setTempLow,0,1);
  if(checkKeyDelay(IDLEDELAY)) return(1);
  mt_printPM(humiditySet,0,0);
  mt_print8BitNumber(setHumidity,0,1);
  if(checkKeyDelay(IDLEDELAY)) return(1);
  if(idleState < 6) idleState++;
  else idleState = 0;
  return(0);
}
```

When the *idle()* function is running, we see the LCD display. The idle mode shows the text on the LCD for a period determined by the *IDLEDELAY* constant used in the *checkKeyDelay* function. In **Figure 3**, we see the flow of the messages with the delay represented by ‘AUTO’ to indicate that the LCD text automatically transitions from one to the next until the user pushes the center button. This will put the application into the menu mode:

```
mt_printPM(tempOut,0,0);
mt_print8BitNumber(tempOutdoor,0,1);
if(checkKeyDelay(IDLEDELAY)) return(1);
```

The *loop()* calls the *idle()* function that then runs through all of the functions to display the idle information. It prints the data name on the first line; the data value on the second line; then it calls *checkKeyDelay(IDLEDELAY)* — the function that checks the key while also delaying for the indicated time, allowing the LCD to display the data for the indicated time. If no key is pressed, then the next set of data is displayed and the buttons checked, and so on until either a key is detected or the data is all displayed. If no key is detected, then the function returns 0 to the loop which runs the *idle()* function again.

If a key is detected, the *idle()* function returns 1 and the loop checks to see if there really is a new key available. If the button was CTR, the program runs the *menu()* function that lets the user input the temperature and humidity settings.

The Menu Mode

One really great thing about the Arduino handheld

prototyper is that it provides the Arduino with an LCD for output and pushbutton keys for input – both accessible via two pins on the I²C bus. You have a library of functions that makes using the LCD and the pushbutton keys a breeze. That said, sometimes the logic for getting user input from the keys and showing data to the user can be a bit of a pain. Not only is the logic tedious, but since you only have two lines with eight characters each and five pushbuttons, you have to think about the best way to show the user what is going on using the minimal characters and keys available.

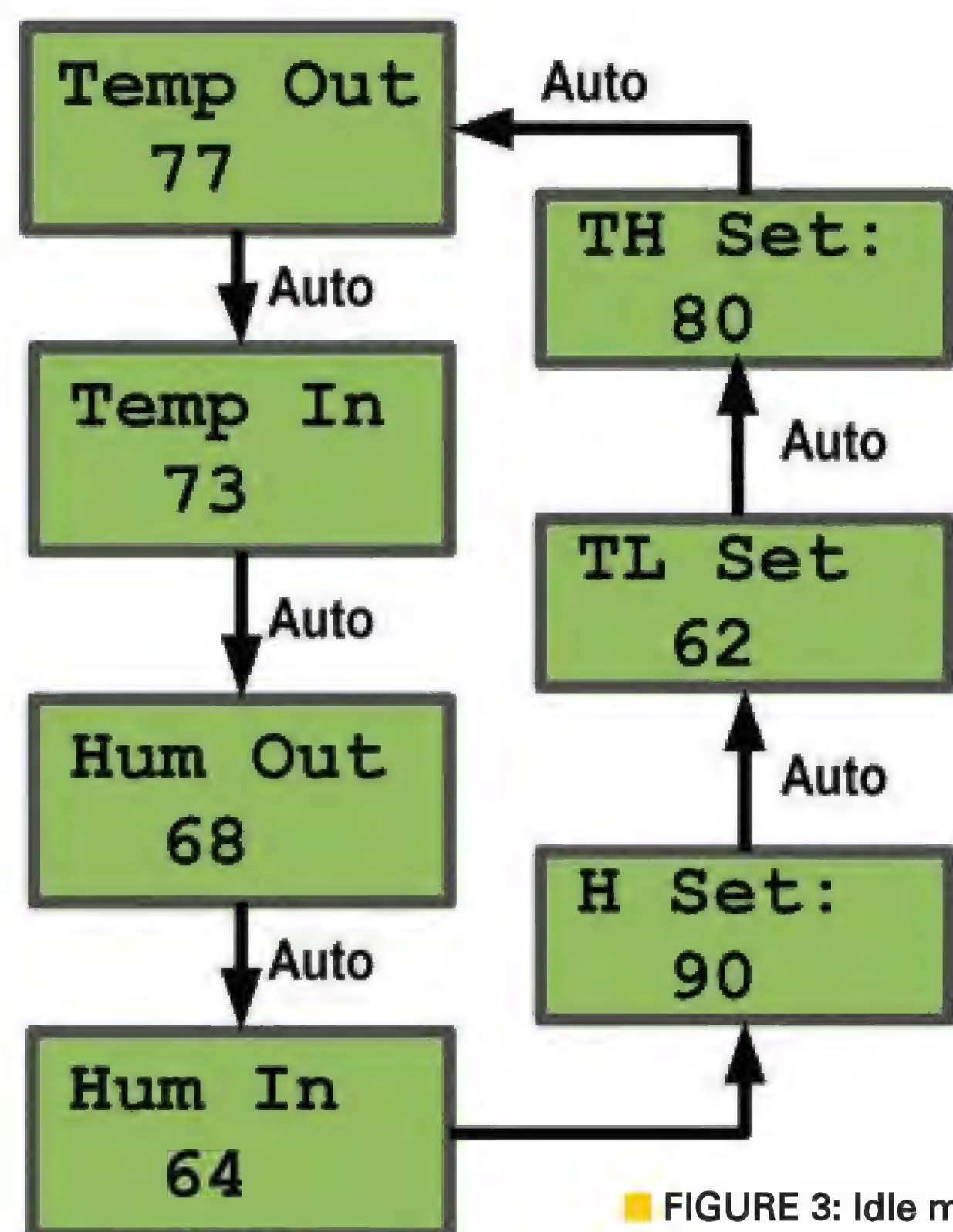
We've all messed with TV remotes and have a good idea of how cryptic these limited systems can sometimes be, but we also know this is an inexpensive way to do things. After all, we could just use a PC if we don't mind the expense.

Typically to use these limited systems, we have some sort of menu structure where the user can use the keys to scroll through text options shown on the LCD and then select the desired action. For the fresh air controller, the actions are fortunately few and simple. We want the user to be able to input the settings for the high and low temperatures and the humidity that the Arduino can compare to the indoor and outdoor sensor readings to decide whether to turn the attic fan on or off. Let's look at the *menu()* function:

```
void menu()
{
    uint8_t inMenu = 1;
    uint8_t key = newKey;

    mt_print("Move Key",0,0);
    mt_print("Up or Dn",0,1);
    delay(500);
    showSelection(0);
    selection = 0;

    while(inMenu)
    {
        if(checkKeyDelay(250))
        {
            checkKey = 0;
            if(newKey)
            {
                if( (newKey == CTR) || (newKey == RGT) )
                // Exit the menu
                {
                    inMenu = 0;
                }
                else if(newKey == UP)
                // Scroll up through menu
                {
                    if(selection > 0)selection--;
                    showSelection(selection);
                }
                else if(newKey == DWN)
                // Scroll down thru menu
                {
                    if(selection == SELECTIONS) return;
                    if(selection < SELECTIONS)
                    selection++;
                    showSelection(selection);
                }
            }
        }
    }
}
```



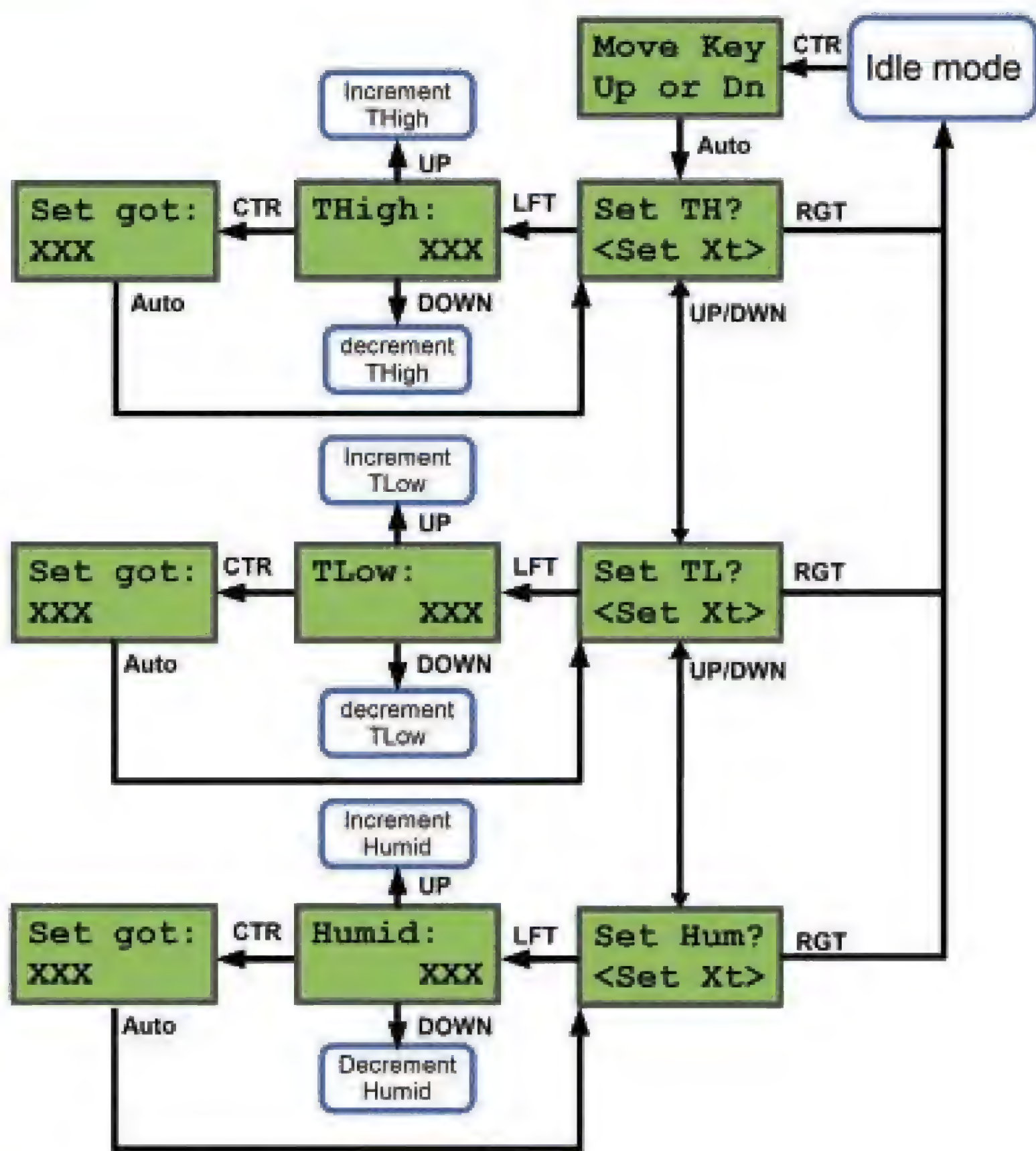
■ FIGURE 3: Idle mode.

```

else if(newKey == LFT)
    // Yes, select the current menu item.
    {
        if(selection == 0) doSetTempHigh();
        else if (selection == 1)doSetTempLow();
        else if (selection == 2)doSetHumidity();
        else // something is wrong
        {
            selection = 0;
            break;
        }
        showSelection(0);
    }
    newKey = 0;
}
}
}
```

When the function first starts, it sets the *inMenu* variable to 1 so the *while* loop will continue to look at keys until it gets either CTR or RGT which will tell it to exit the function. It then takes a second to tell the user to move the keys up or down. Then, it moves to the 'Set TH' menu item. This is all illustrated in **Figure 4**.

Each menu image is shown in an LCD-like box with arrows showing what happens when you press a key. The 'Auto' arrow indicates that the action happens automatically. So, for instance, when you set the high temperature and click CTR, you will see the temperature



■ FIGURE 4: Menu mode.

you set on the LCD. After a moment, the menu returns to the 'Set TH?' item where it waits for a button press.

From that menu item, if you press the rightmost button the software returns to the Idle mode. If you press the left key, it goes to the 'Thigh:' menu item where you can set the value for the high temperature. If you press the down key, the menu goes to the 'Set TL?' (Set Temperature Low) menu item.

This all reads rather complicated, so it is best to do the 'picture is worth a thousand words' thing and refer to **Figure 4** to see what is really going on. A good exercise is to look at the **figure** and at the code to see how I've implemented this particular menu. My coding style is meant to be educational and may not be the most efficient way to do this, but it is the clearest to me. This should help you understand the style of menu design and programming so that you can follow the method and create your own menus.

The menu calls the *showSelection* function to write the indicated selection to the LCD:

```
void showSelection(uint8_t selected)
{
    // Show selected query
    if(selected == SETTH)
    {
```

```
        mt_printPM(setTHigh,0,0);
    }
    else if (selected == SETTL)
    {
        mt_printPM(setTLow,0,0);
    }
    else if (selected == SETHUM)
    {
        mt_printPM(setHumid,0,0);
    }
    else // something is wrong
    {
        mt_print("????????",0,0);
        delay(1000);
        return;
    }

    // See if the user wants to set this
    // value
    mt_printPM(yes,0,1);
}
```

Notice this has nothing to do with what the Arduino does with the selected item, other than display the selection on the LCD. The work indicated by the selection is done in the menu function under the LFT key part. There – depending on the selection – the working function is called. For example, the *doSetHighTemperature()* function follows:

```
void doSetTempHigh()
{
    setTempHigh = mt_get8BitNumber
    ("THigh:",0);

    mt_print("STH got:",0,0);
    mt_print8BitNumber(setTempHigh,0,1);
    delay(1000);
}
```

In this function, the I²C mini terminal is instructed to get an eight-bit number using the second line, and to print "Thigh:" in the first line. The returned value is placed in the *setTempHigh* variable; then, the LCD is told to print the value it got, wait a second, and return. Again, this is a good time to look at **Figure 4** and follow the code to see what is happening.

How It Controls the Fan

Last month, we looked at the algorithm used to turn the fan on or off based on the indoor and outdoor temperature and humidity, based on user input set points. The Idle routine checks the sensors periodically by calling the *checkSensors()* function we saw last month, then based on the results of the algorithm sets the *fanState* variable to either 0 or 1. This value is used in the *setFan()* function that follows:

```
// Turn the fan on or off
void setFan(uint8_t state)
{
    if(fanState == 1)// turn it on
    {
```



```

    digitalWrite(fanPin,HIGH);
    // Set pin to Vcc
}
else // turn it off
{
    digitalWrite(fanPin,LOW);
    // Set pin to ground
}
}

```

This results in the *fanPin* being set either high (+5) or low (GND). As you can well imagine, the Arduino isn't able to supply much +5 volt current to actually run a fan. You have to use this pin to signal an external device — some sort of transistor and/or relay arrangement that turns the fan on and off. Since the fan is almost certainly being powered by mains AC current, you must really know what you are doing to accomplish that step without killing somebody or burning down your castle. My first recommendation is to thoroughly test your system by turning an LED (a fan substitute) on and off to verify that it is all working properly.

As of this moment, that's all I've got on operating the fan. Next month — assuming everything is working properly — we should have the system fully hooked up and tested. Then, we'll return to the dangerous stuff. In the meantime, wouldn't it be useful to record all this data we are collecting on the temperature and the humidity, and when we are turning the fan on and off?

Well, first we will want to know the 'when' part by teaching our fresh air controller how to tell time.

Teaching It to Tell Time

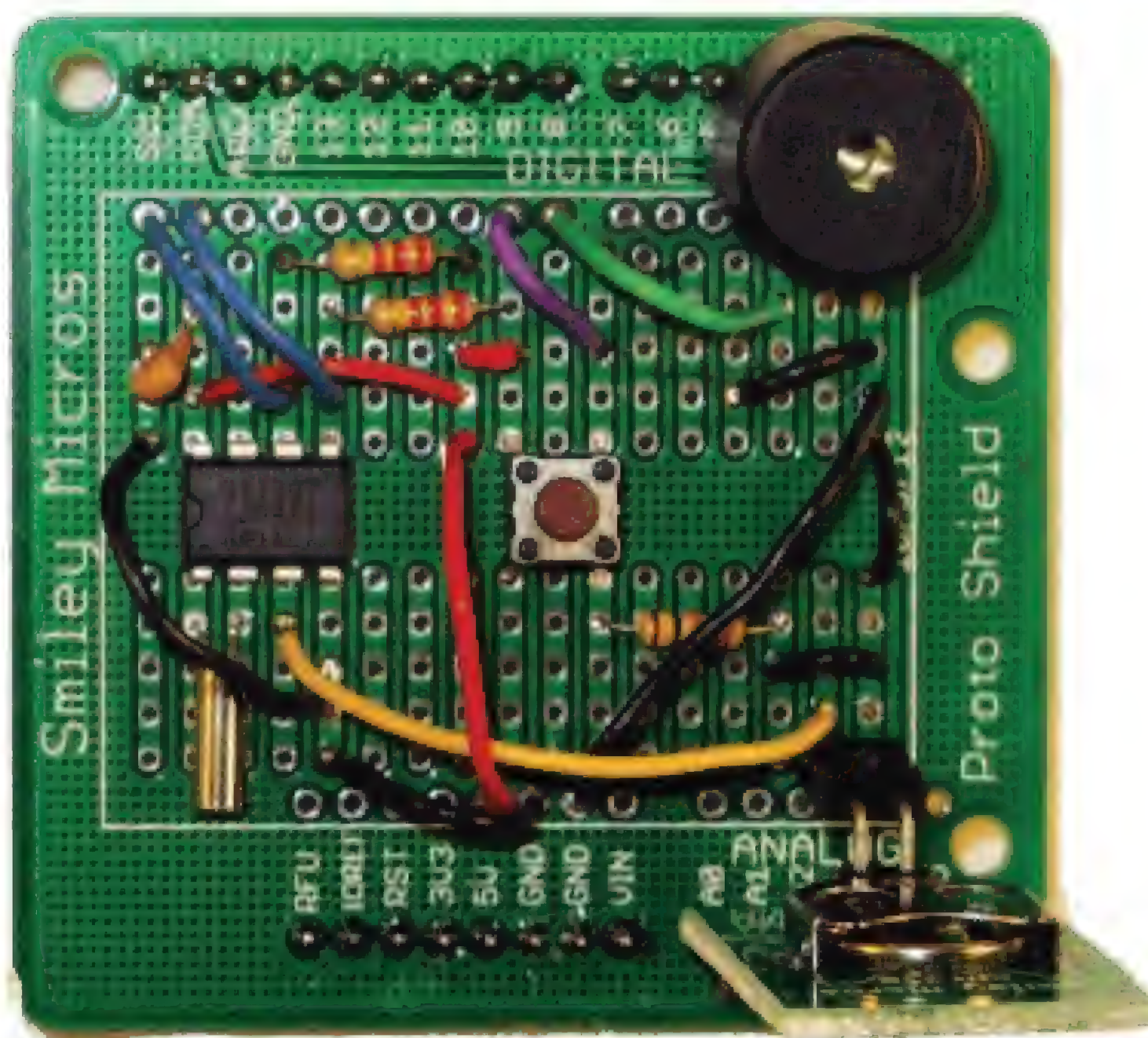
Fortunately, we just recently went through a bunch of Workshops where we learned a lot about how to keep dates and time on a computer. This began with the January 2013 Workshop where we started looking at how to use the Arduino proto shield to design and build an alarm clock. We then continued through the April Workshop. You can find excerpts of these on my articles repository at blog.smileymicros.com. You can also purchase the associated Arduino proto shield and/or the proto shield alarm clock kit from the *Nuts & Volts* webstore.

For this section, I will assume you've already reviewed that material and can refer back to it if you have questions.

So, Where Do We Put the Clock?

Good question. The fresh air controller pretty much fills out the Arduino proto shield, so where do we put it? Well, remember those stacking headers? Right! We will stack them. **Figure 5** shows the alarm clock circuit built on a proto shield PCB (printed circuit board).

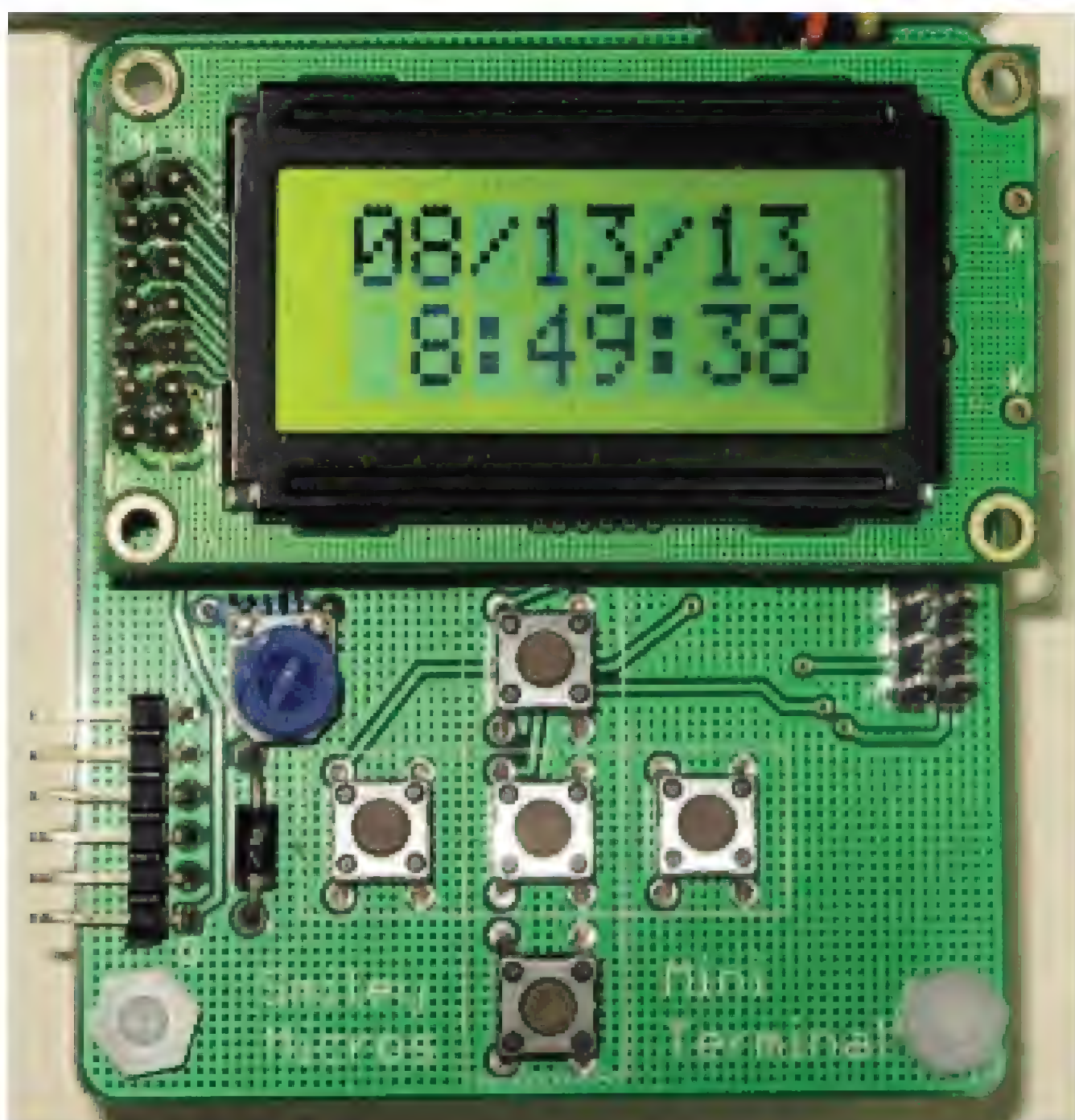
For the fresh air controller board, you'll need to add connectors for the DHT22 temperature and humidity



■ FIGURE 5: Arduino alarm clock.



■ FIGURE 6: Fresh air controller board with alarm clock.



■ FIGURE 7: Date and time on LCD.

sensors so they will have long wires and can be placed off the board. This will make room to stack the alarm clock on top as shown in **Figure 6**.

Showing Dates and Times On the LCD

Eight characters by two lines doesn't give us a lot of space to show dates and times, but it is all we actually need. Both dates and times are traditionally shown with three two-digit numbers separated by two characters: '/' for dates and ':' for time — meaning that eight characters is exactly what we need to show them as we can see in **Figure 7**.

In our earlier Workshops, we saw how to get the date and time from the DS1307 in the *DateTime* data format. Now, we'll see how to convert that into characters we can show on the LCD (**Figure 7**). As usual, there are a couple of unexpected complexities.

The *DateTime* data type is in military time; that is, hours go from 0 to 23 which is fine if we want to display the time like that, but not if we want to do it the typical way a clock is displayed. We define the variable *militaryTime* that we set to 1 if we want to display military time, and to 0 if we want the regular time display.

The next complexity is that when we use the *itoa()* (integer to ASCII) function, we get a buffer with a single character for date and time values less than 10, and two characters for values 10 and above. We have to compensate for this when we display the values, adding a leading 0 to those values from 1 to 9. The following

timeToChar function shows how this is done for time. For dates, we use the *dateToChar* function that is identical in logic, so we won't show it here:

```
void timeToChar(char * tim, DateTime myDateTime)
{
    char buff[4];

    int hour = myDateTime.hour();

    if( !militaryTime && (hour >= 12)) hour -= 12;

    // convert the hour
    itoa(hour,buff,10);
    if(hour >= 10)
    {
        tim[0] = buff[0];
        tim[1] = buff[1];
    }
    else
    {
        tim[0] = ' ';
        tim[1] = buff[0];
    }
    tim[2] = ':';
    // Convert the minute
    int min = myDateTime.minute();
    itoa(min,buff,10);
    if(min >= 10)
    {
        tim[3] = buff[0];
        tim[4] = buff[1];
    }
    else
    {
        tim[3] = '0';
        tim[4] = buff[0];
    }
    tim[5] = ':';
    // Convert the second
    int sec = myDateTime.second();
    itoa(sec,buff,10);
    if(sec >= 10)
    {
        tim[6] = buff[0];
        tim[7] = buff[1];
    }
    else
    {
        tim[6] = '0';
        tim[7] = buff[0];
    }

    // make the char array a string by terminating
    // it with the NULL '\0' character
    tim[8] = '\0';
}
```

The software to use the Arduino alarm clock proto shield kit with the I²C mini terminal LCD is in the Arduino sketch *textDateTimeLCD* which is available along with the most recent revision of the fresh air controller at the [article link](#).

Next month, we will put the date and time code into the fresh air controller so that we can see it as part of the *idle()* function. We will also begin to learn how to record the data and report it out to a PC so that we can graph the results. **NV**

Motor Controllers • Arduino • Signal Generators • FM Transmitters • Timers • Audio Amplifiers

200 KHz Function Generator

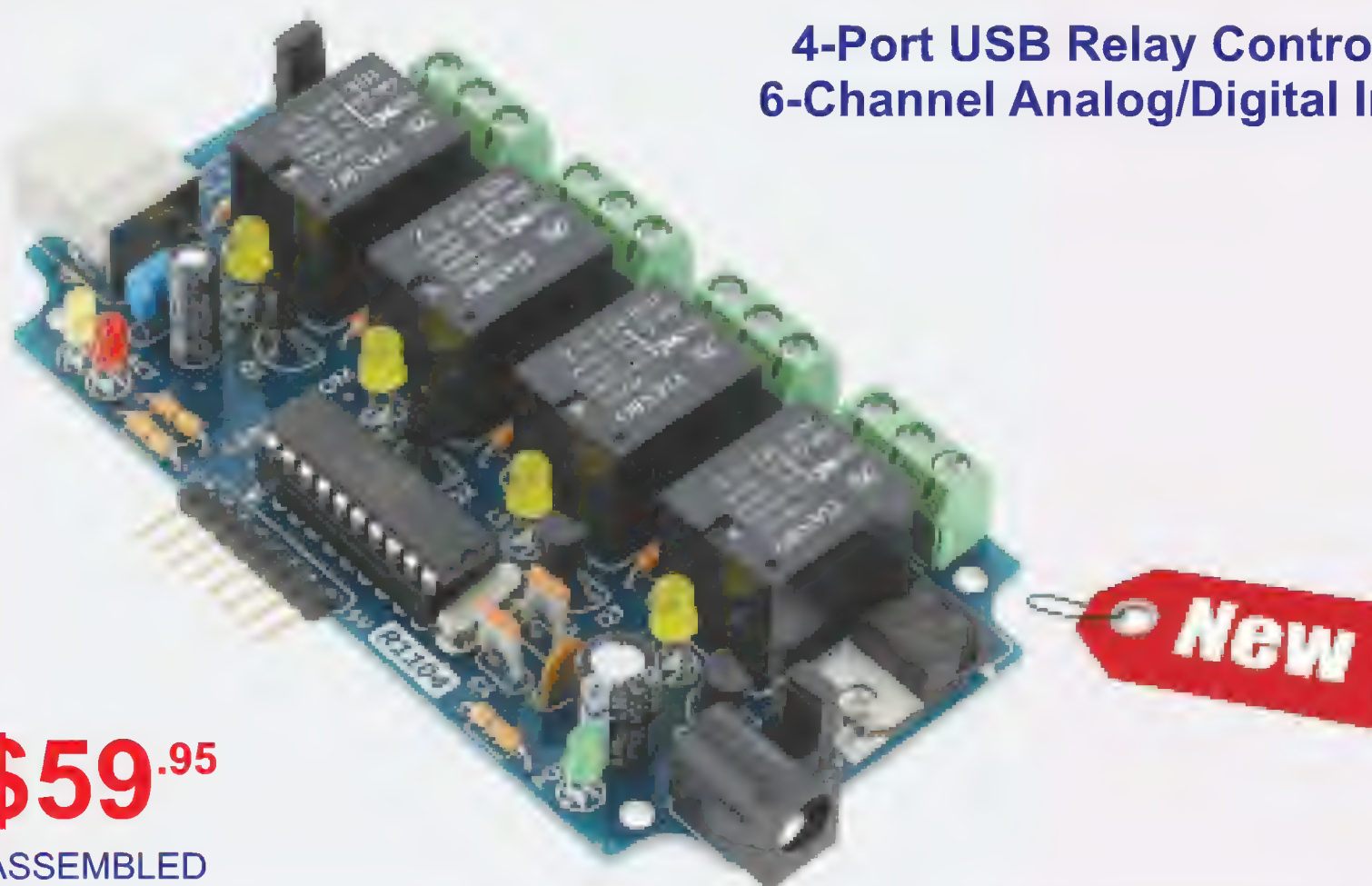


\$49^{.95}

ASSEMBLED

UK307

4-Port USB Relay Controller with
6-Channel Analog/Digital Interface



\$59^{.95}

ASSEMBLED

UK1104

5A Motor Speed Controller (PWM)

\$22^{.95}

ASSEMBLED



UK1102

Stereo Audio Amplifier (10W)



\$35^{.95}

ASSEMBLED

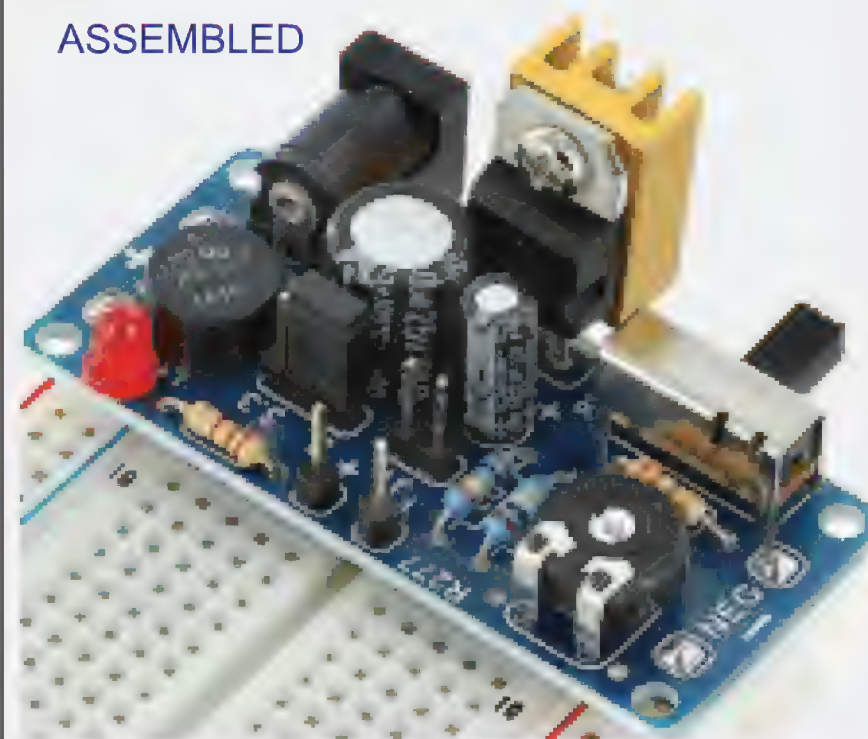
UK560

Plug-In Breadboard Power Supply

\$19^{.95}

ASSEMBLED

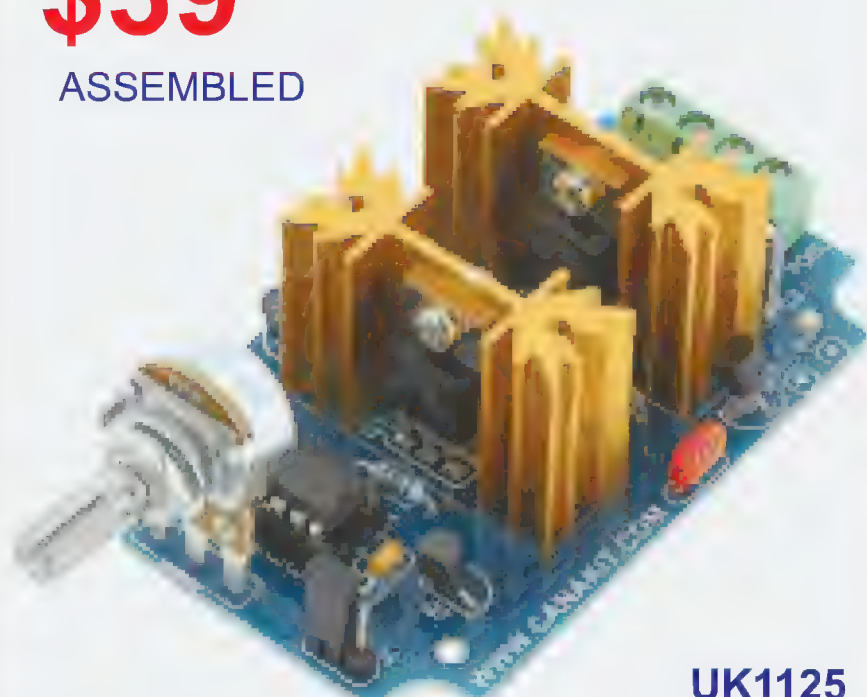
UK277



7A Bi-Directional
Motor Speed Controller (PWM)

\$39^{.95}

ASSEMBLED



UK1125

USB PIC Programmer
(MPLAB Compatible)



\$69^{.95}

ASSEMBLED

UK1300

Digital LED Timer (Up/Down)

UK258



\$46^{.95}

ASSEMBLED

For complete product details, visit our webstore!!

The Nuts & Volts **WEBSTORE**

GREAT FOR DIYers!

Getting Started with chipKIT by Chuck Hellebuyck

In this book, Chuck Hellebuyck — who has authored many entry level technical books — shows you how to get started with the chipKIT Uno32 using some very simple example sketches (sketches are a software program in the Arduino world) that demonstrate how to use digital inputs, digital outputs, analog inputs, and analog outputs. These are the fundamental building blocks every electronic project needs. With the examples in this book, you'll have what you need to get your electronic project up and running.

\$19.95

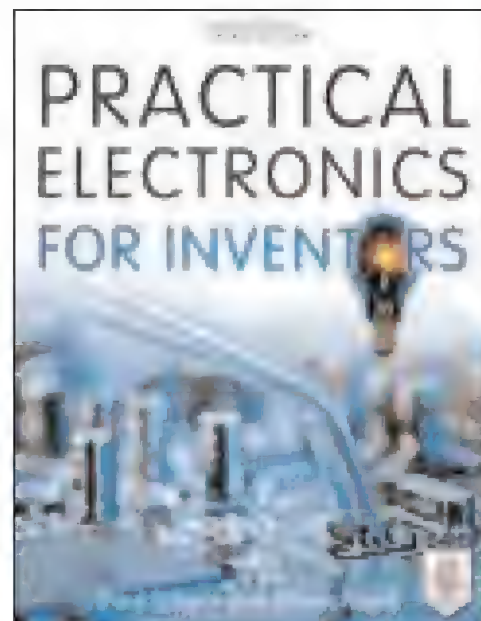


Practical Electronics for Inventors, Third Edition by Paul Scherz, Simon Monk

The electronics know-how you need to become a successful inventor.

Spark your creativity and gain the electronics skills required to transform your innovative ideas into functioning gadgets. This hands-on updated guide outlines electrical principles and provides thorough easy-to-follow instructions, schematics, and illustrations.

\$29.95

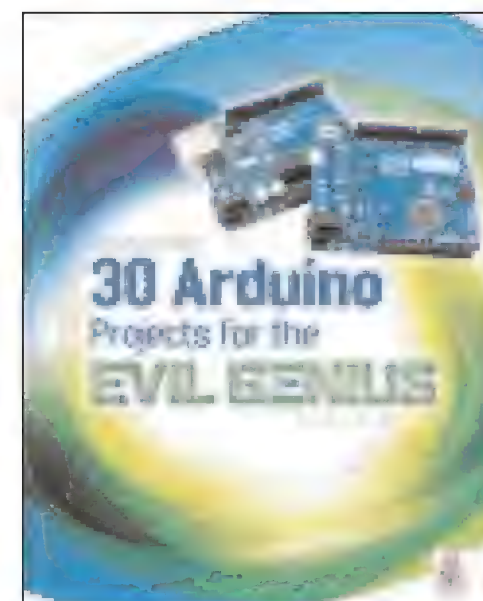


NEW!

30 Arduino Projects for the Evil Genius: Second Edition by Simon Monk

Fully updated throughout, this do-it-yourself guide shows you how to program and build fascinating projects with the Arduino Uno and Leonardo boards, and the Arduino 1.0 development environment. *30 Arduino Projects for the Evil Genius, Second Edition*, gets you started right away with the simplified C programming you need to know, and demonstrates how to take advantage of the latest Arduino capabilities.

\$19.95



Build Your Own Transistor Radios by Ronald Quan A Hobbyist's Guide to High Performance and Low-Powered Radio Circuits

Create sophisticated transistor radios that are inexpensive yet highly efficient. Inside this book, it offers complete projects with detailed schematics and insights on how the radios were designed. Learn how to choose components, construct the different types of radios, and troubleshoot your work.

***Paperback, 496 pages
\$49.95**

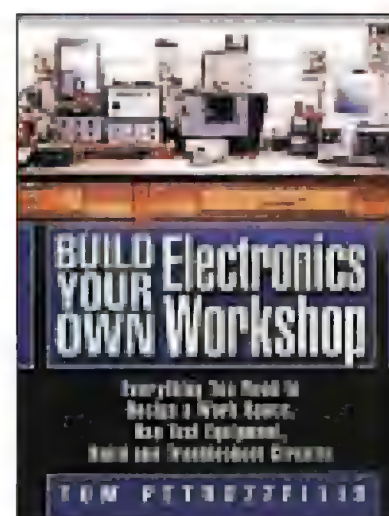


Build Your Own Electronics Workshop by Thomas Petruzzellis

BUILD YOUR OWN DREAM ELECTRONICS LAB!

This value-packed resource provides everything needed to put together a fully functioning home electronics workshop! From finding space to stocking it with components to putting the shop into action — building, testing, and troubleshooting systems. This great book has it all! And the best part is, it shows you how to build many pieces of equipment yourself and save money, big time!

**Reg Price \$29.95
Sale Price \$26.95**



Beginner's Guide to ... Programming the PIC24/dsPIC33 by Thomas Kibalo

Kibalo takes you step by step through the fundamentals of programming the PIC24H which can equally be applied to the dsPIC33. His clear explanation of the inner workings make learning the PIC24H/dsPIC33

16-bit architecture easy. His code examples demonstrate how to perform the functions most applications require. The hardware is shown in a simple breadboard setup so even a beginner can build it, along with very few extra components needed.

\$39.95*

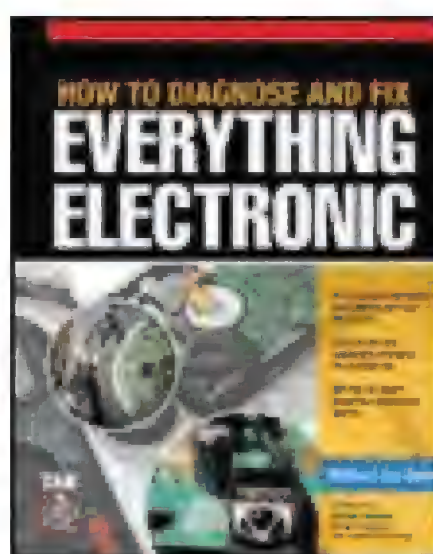


How to Diagnose and Fix Everything Electronic by Michael Jay Geier

Master the Art of Electronics Repair

In this hands-on guide, a lifelong electronics repair guru shares his tested techniques and invaluable insights. *How to Diagnose and Fix Everything Electronic* shows you how to repair and extend the life of all kinds of solid-state devices, from modern digital gadgetry to cherished analog products of yesteryear.

\$24.95



Programming PICs in Basic by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you've found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller. **\$14.95**



Master and Command C for PIC MCUs by Fred Eady

Master and Command C for PIC MCU, Volume 1 aims to help readers get the most out of the Custom Computer Services C compiler for PIC microcontrollers.

The author describes some basic compiler operations that will help programmers particularly those new to the craft create solid code that lends itself to easy debugging and testing. As Eady notes in his preface, a single built-in CCS compiler call (output_bit) can serve as a basic aid to let programmers know about the "health" of their PIC code. **\$14.95**



Order online @ www.store.nutsvolts.com
Or CALL 1-800-783-4624 today!

BOOK & KIT COMBOS

16-Bit Complete Combo



Combo Price \$319.95

For complete details visit our webstore @ www.nutsvolts.com

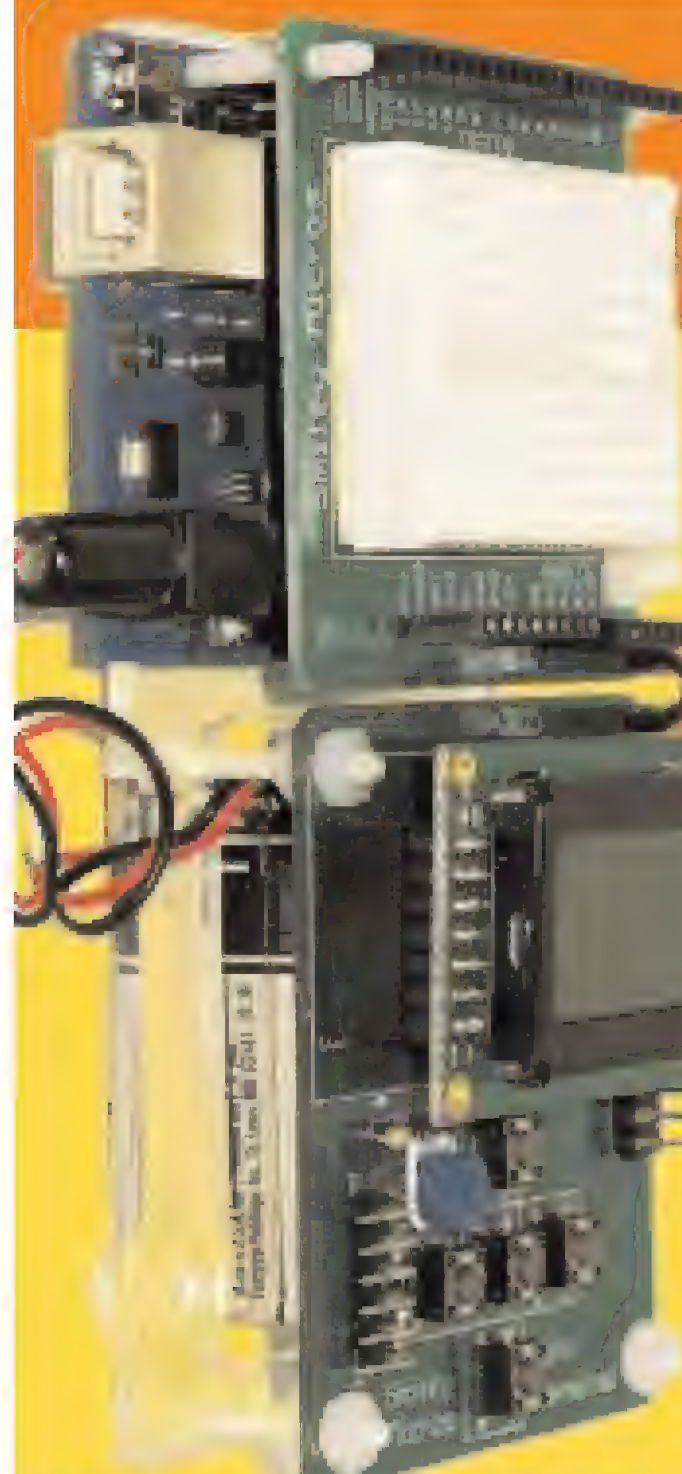
Beginner's Guide Complete Combo!



Combo Price
\$229.95

For complete details, visit our webstore @ www.nutsvolts.com

Arduino Handheld Prototyper System



Arduino Proto Shield
\$16.99

I2C Mini Terminal
\$29.99

Handheld Base
\$13.99

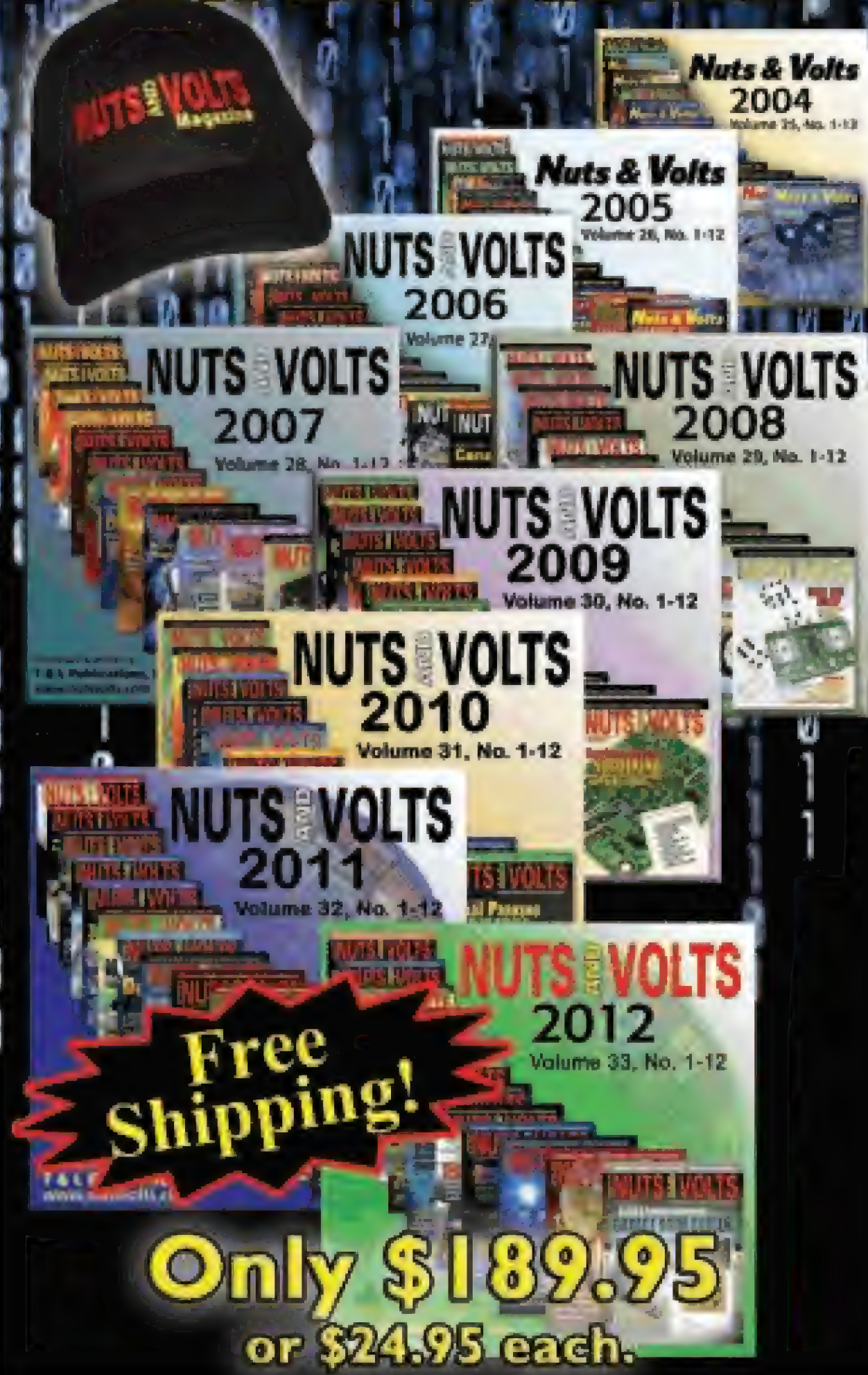
All three*: \$59.99

*Arduino and battery not included

From Smiley's Workshop

CD-ROM SPECIAL

**Nuts & Volts
9 CD-ROMs
& Hat Special!**
That's 108 issues.
Complete with supporting
code and media files.



Only \$189.95
or \$24.95 each.



The Nuts & Volts Pocket Ref

All the info you need at your fingertips!

This great little book is a concise
all-purpose reference featuring
hundreds of tables, maps,
formulas, constants &
conversions.
AND it still fits in
your shirt pocket!

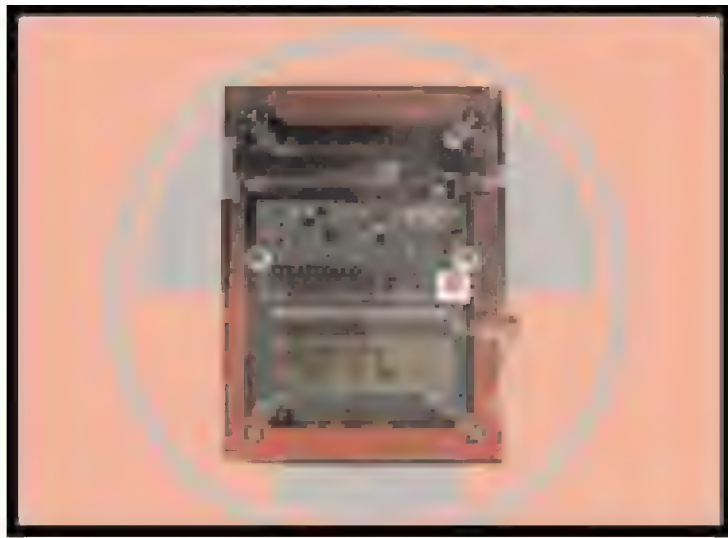
Only \$12.95

Visit <http://store.nutsvolts.com> or call (800) 783-4624

CALL 1-800-783-4624 today!
Order online @ www.nutsvolts.com

PROJECTS

Geiger Counter Kit



This kit is a great project for high school and university students. The unit detects and displays levels of radiation, and can detect and display dosage levels as low as one micro-roentgen/hr. The LND712 tube in our kit is capable of measuring alpha, beta, and gamma particles.

Subscriber's Price \$145.95
Non-Subscriber's Price \$148.95

3D LED Cube Kit



This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes. Colors available: Green, Red, Yellow & Blue

Subscriber's Price \$57.95
Non-Subscriber's Price \$59.95

Radiation Monitor Alarm Kit



This is an inexpensive surface-mount project that is good for beginners to start with. This kit has its own printed circuit board (PCB) which makes mounting the components easy. Plus, it comes in a pocket size shielded aluminum case measuring 3.5" in length and 1" in diameter. The on-off switch is a pushbutton on the bottom.

Subscriber's Price \$32.45
Non-Subscriber's Price \$33.95

Neon Transistor Clock Kit



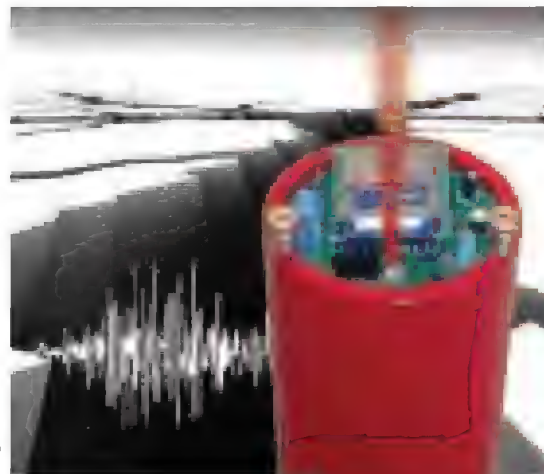
Add HIGH VOLTAGE to your clock!
 This is a Nixie Tube display version of the Transistor Clock. It uses only discrete components — no integrated circuits. For more info, see the April 2012 issue.

Subscriber's Price \$245.95
Non-Subscriber's Price \$249.95

Seismograph Kit

As seen in the May 2012 issue.

Now you can record your own shaking, rattling, and rolling.



The Poor Man's Seismograph is a great project /device to record any movement in an area where you normally shouldn't have any. The kit includes everything needed to build the seismograph. All you need is your PC, SD card, and to download the free software to view the seismic event graph.

Subscriber's Price \$79.95
Non-Subscriber's Price \$84.95

Magic Box Kit

As seen in the November 2011 issue.

Battery Marvel helps protect cars, trucks, motorcycles, boats, and any

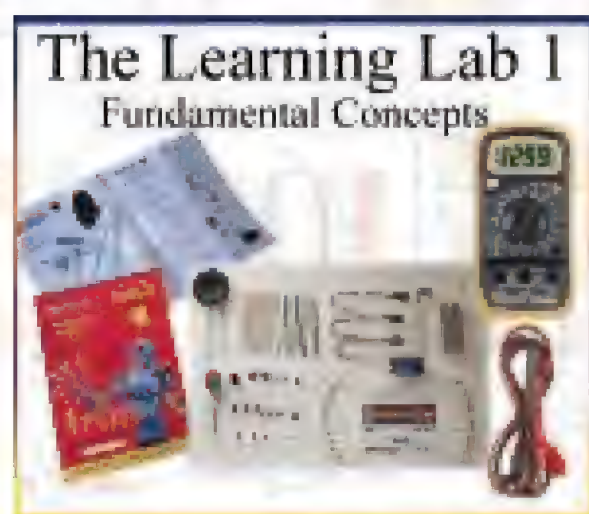


other 12V vehicles from sudden battery failure. This easy-to-build kit features a single LED that glows green, yellow, or red, indicating battery health at a glance. An extra-loud piezo driver alerts you to any problems.

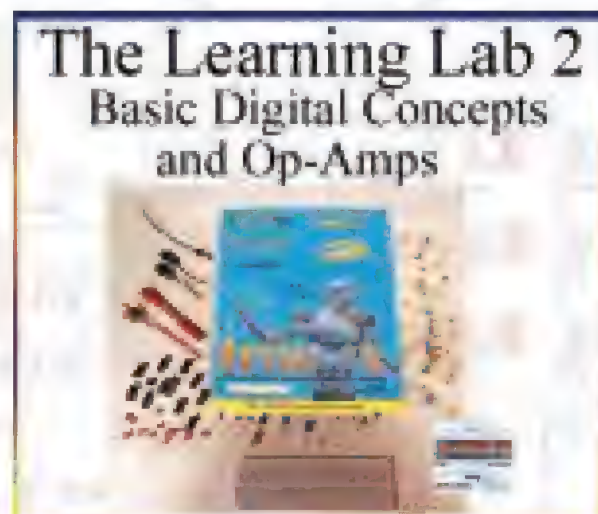
Details, please visit our website.

Subscriber's Price \$18.95
Non-Subscriber's Price \$19.95

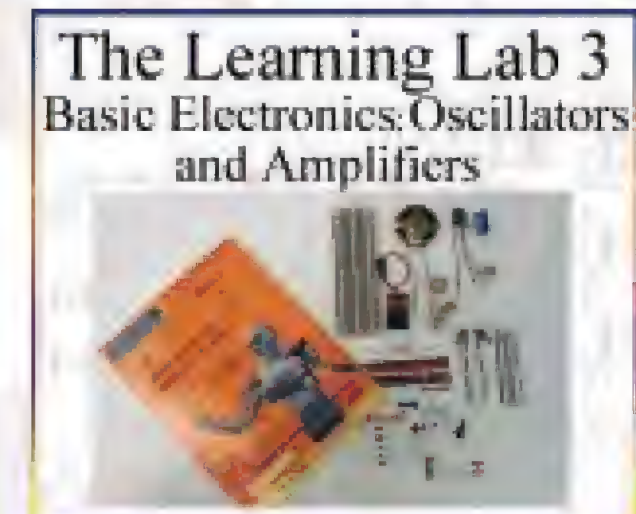
FOR BEGINNER GEEKS!



\$59.95



\$49.95



\$39.95

The labs in this series — from GSS Tech Ed — show simple and interesting experiments and lessons, all done on a solderless circuit board.

As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal of knowledge with each successive experiment.

For more info and a promotional video, please visit our webstore.

ELECTRONET

GET THE NUTS & VOLTS DISCOUNT!
Mention or enter coupon code **NVRMZ12**
and receive 10% off your order!

ramsey
www.ramseykits.com
AM/FM Broadcasters • Hobby Kits
Learning Kits • Test Equipment
...AND LOTS OF NEAT STUFF!



www.canakit.com

- Over 200 Kits & Modules to choose from!
- Easy Worldwide Online Ordering
- Fast Delivery & Same Day Shipping

FM Transmitters • Timers • Audio Amplifiers • Motor Controllers



THE ORIGINAL SINCE 1994
PCB-POOL
Beta LAYOUT

- Low Cost PCB prototypes
- Free laser SMT stencil with all Proto orders

WWW.PCB-POOL.COM

Serial LCD & LED
www.melabs.com



Got electronics?
www.nutsvolts.com

ALL ELECTRONICS
CORPORATION
Electronic Parts & Supplies
Since 1967



For the ElectroNet
online, go to
www.nutsvolts.com
click **Electro-Net**

INVEST in your BOT!



INVEST in **HITEC**

12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecusa.com

d6 LABS

Official US Distributor
for **HopeRF** Modules

www.d6labs.com

RF Modules ■ Antennas ■ Connectors

USB Add USB to your next project--
It's easier than you might think!

USB-FIFO • USB-UART • USB/Microcontroller Boards
RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!

www.dlpdesign.com



superbrightleds.com

COMPONENT LEDs • LED BULBS • LED ACCENT LIGHTS



Aztec
MCU Prototyping

PICAXE and Arduino design and development tools
www.aztecmcu.com



FREE PCB Design Software

Get Started Now
at **FabStream.com**



Karlsson Robotics

For Electronic Components, SMT Adapters,
Modules, Custom Design, Assembly and Prototyping.

www.kr4.us



CORIDIUM

ARM controllers running
floating point **BASIC** from \$10.00

www.coridium.us

Pololu
Robotics & Electronics
WWW.POLOLU.COM



10 Things You Need to Know About Wi-Fi

Some facts about the familiar wireless technology that you may not know.

Wi-Fi is that popular short-range wireless technology that practically all of us use each day. Wi-Fi is the trade name of the Institute of Electrical and Electronics Engineers (IEEE) 802.11 wireless local area network (WLAN) standard that is widely used to connect PCs, laptops, tablets, cell phones, and other things to the Internet and other services. The Wi-Fi Alliance (WFA) is the professional organization that promotes Wi-Fi, develops enhancements, and implements testing and certification of products to ensure interoperability.

Wi-Fi uses the unlicensed industrial-scientific-medical (ISM) bands of 2.4 and 5 GHz. You probably use the most recent variation of Wi-Fi called 802.11n or at least the previous generation 802.11g to connect to your home network, a nearby hot spot, or company LAN access point.

Here are some interesting variations of the standard and some things you can expect to see in the future.

#1 — The next generation of Wi-Fi ready to go

Every several years, the IEEE standard gets updated to create the next-generation technology. It is always an improvement in some way over the previous version. The next version of the standard is 802.11ac or just 11ac. It is now just beginning to appear in new routers, gateways, computers, and other devices. It is a much faster version than the current 11n. Its primary benefit is much faster data transfers, making it more convenient than ever to download or stream video.

The 11ac standard uses the 5 GHz ISM band. There is less interference in this band, although the range is more limited than in the 2.4 GHz band. There is plenty of spectrum there to allow each data channel to be much wider than the current 20 MHz or 40 MHz. With 80 or 160 MHz channels, data speeds can easily exceed 1 Gb/s or faster under the right conditions.

Modulation is still OFDM but adds up to 256QAM for extra speed.

A typical single stream of data can be as much as 910 Mb/s. Using multiple input multiple output (MIMO) up to 8x8, data rates can peak at 3 Gb/s.

MIMO is the use of multiple transmitters, receivers, and antennas transmitting different data streams in parallel and concurrently to multiply the data rate while extending the range.

The 11ac routers are just becoming available and a few laptops now offer 11ac capability. In the coming year, 11ac will grow significantly. All the new routers will be backwards compatible to support the older standards.

#2 — An even faster version is in the wings

A newer standard called 802.11ad (or just 11ad) is under development and will soon find its way into more products. While not technically Wi-Fi, it is a good complement to the current versions. The 11ad standard is referred to as WiGig. It uses the 60 GHz ISM band (57-64 GHz). While the communications range is short – usually less than 10 meters – it is super fast with data rates as high as 7 Gb/s. Its application will be primarily for video transport from set top box to TV set, DVD to TV, or laptop to docking station. It uses MIMO as well as antenna beam forming and tracking to boost transmit power and extend the range of the system. Some 11ad products are available, but soon others will join the club and the applications will spread.

11ad is supported by the Wireless Gigabit Alliance (WiGig) –

MIMO is the use of multiple transmitters, receivers, and antennas transmitting different data streams in parallel and concurrently to multiply the data rate while extending the range.

between devices — again without going through the router. Miracast-enabled devices like a laptop can send its video directly to a big screen HDTV, or a laptop can transfer its video to a docking station with a larger video monitor. It can be used as the physical layer for popular wired interface standards

like USB, PCIe, HDMI, and DisplayPort. Miracast is available now in many products.

#5 — HotSpot 2.0 makes Wi-Fi easier to use and connect

HotSpot 2.0 — also known as Passpoint — is one more WFA standard that facilitates and simplifies finding hotspots nearby and connecting to them automatically (or with minimal user interaction). What makes this standard possible is another IEEE standard: 802.11u. This software addition lets devices and networks negotiate with one another and make automatic connections.

#6 — Small cells complement the cellular system

Wi-Fi will soon act somewhat like a cellular base station or cell site. This feature using HotSpot 2.0 is seen as the future because it will enable cell phone carriers to use Wi-Fi to offload data from their 4G networks. The 4G networks are busy and often overloaded. The solution to this problem (and higher download speeds) is to add many more

miniature base stations called small cells that are part of a larger heterogeneous network (HetNet). In the meantime as 4G rollouts continue, Wi-Fi offload will let the carriers shift any big data transfers from the 4G networks to any available Wi-Fi networks. This lightens the burden on the cellular network until small cells come along, and simultaneously makes data transfers faster for the user. Wi-Fi offload is not widely implemented yet, but HotSpot 2.0 is now available in some devices.

#7 — Activate security

Some surveys have shown that the majority of home Wi-Fi networks do not have the security function activated. That means that anyone within range of your router (less than about 300 feet) can access your network and connect to whatever — whether you know it or not. Not good! Wi-Fi does provide excellent security in the form of several types of encryption, however. The original security system called wired equivalent privacy (WEP) has been shown to be too weak, so more elaborate systems were created under the IEEE 802.11i standard. The most recent and best is WPA2, or Wi-Fi Protected Access 2. To ensure maximum privacy and security, turn on WPA2. You have to enter a password to get in but it does guarantee that you are the only user.

#8 — Your smartphone as a hotspot

Most smartphones can be used as a hotspot to provide connectivity for

an organization that works with the Wi-Fi Alliance to ensure good compatibility between 11ac and 11ad. Look for tri-band routers with 2.4, 5, and 60 GHz capability.

Table 1 is an overview of the current standards and their characteristics.

#3 — Wi-Fi Direct is now available

Wi-Fi Direct is a WFA software variation that allows two Wi-Fi enabled products to talk to one another directly without going through a nearby access point. This allows PCs and laptops to communicate directly with printers, cameras, TV sets, and other devices, bypassing the router. Tablets and smartphones can talk directly to other devices, as well. Most new devices are already Wi-Fi Direct capable.

#4 — Miracast facilitates video transfer

This is another WFA standard that permits the transfer of video directly

Table 1 Summary of current 802.11 Wi-Fi standards.

IEEE Std	Ratification Date	Band	Technology	Modulation	Channel BW	MIMO (Antennas)	Max Speed
802.11n	2009	2.4 & 5 GHz	OFDM	Up to 64QAM	20, 40 MHz	Up to 4x4	600 Mb/s
802.11ac	2013-14	5 GHz	OFDM	Up to 256QAM	40, 80, 160 MHz	Up to 8x8	3 Gb/s
802.11ad	2012	60 GHz	OFDM	Up to 256QAM	2.16 GHz	Beam forming	7 Gb/s

Wi-Fi enabled devices. Your smartphone connects to the Internet via your cellular provider. Then, any Wi-Fi device — like a laptop, tablet, or other smartphone — can link to your smartphone. Your phone serves as the wireless router. You will need to determine if your phone will do this and what restrictions (if any) your carrier requires.

#9 — White space may use Wi-Fi

White space is the term used to refer to the unused TV channels around the country. The goal is to use these vacant channels as ways for homes

and businesses to connect to the Internet as an option to other high speed broadband services from cable TV or phone company DSL. The location of these blank channels varies by locale, so a method has been developed to locate each vacant channel. A database of these channels and other wireless devices (like wireless microphones) has been established, and any white space data radio has to access the database to find a vacant channel that will be the least likely to cause interference.

Most of these channels are UHF TV channels 20 to 31 in the 470 to 710 MHz range. Each channel is 6 MHz wide so can carry high data

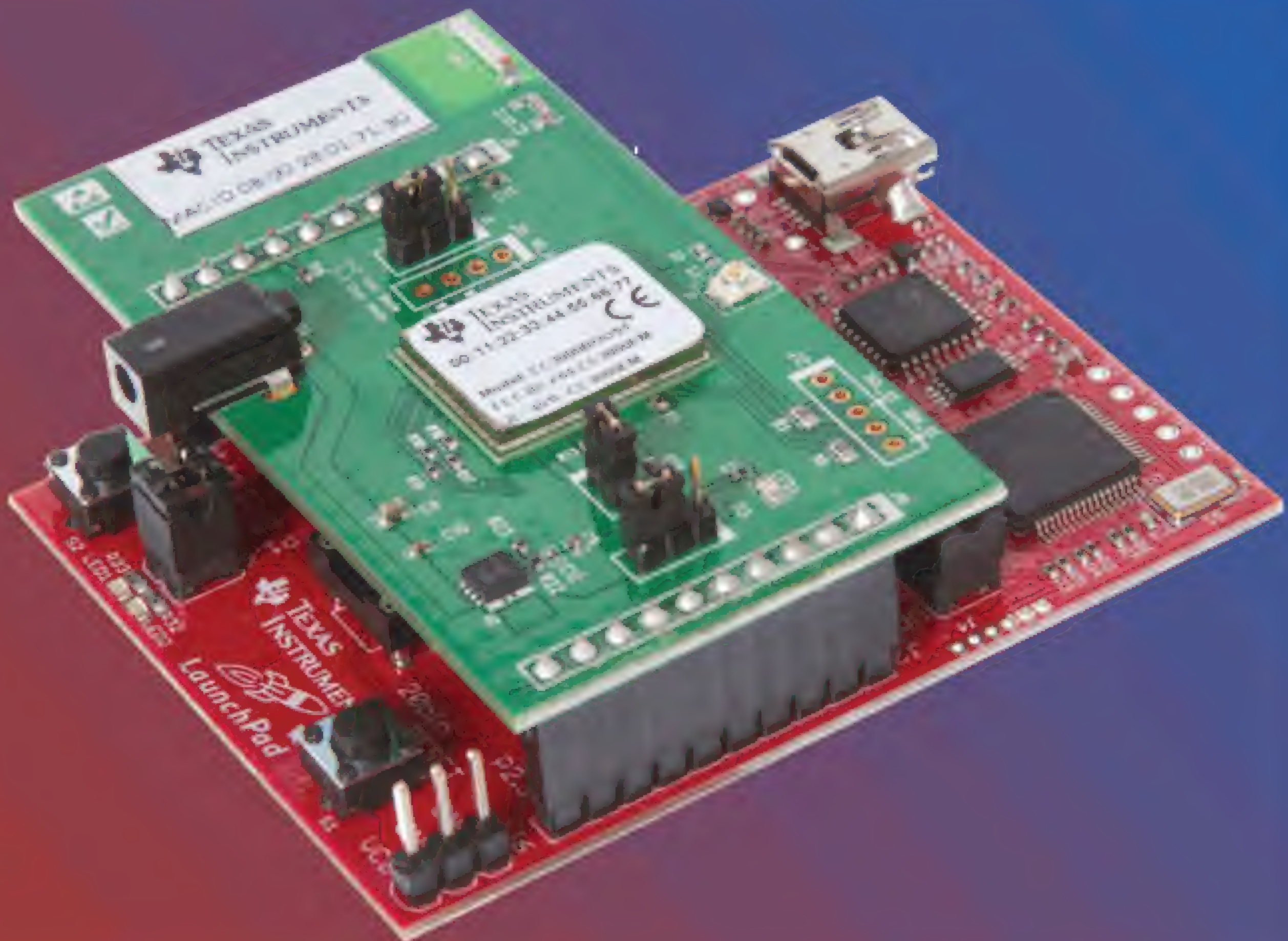
rates with the right modulation. A special version of Wi-Fi — which many refer to as Super Wi-Fi — is based on the IEEE 802.11af standard. It uses OFDM and up to 64QAM to provide data rates from a few Mb/s to 20 Mb/s or so, depending on range and noise conditions. The range at these UHF bands greatly exceeds the range of most 2.4 and 5 GHz versions of Wi-Fi, which is 100 meters at best and usually less. White space range can be up to several miles under the right conditions.

White space radios and services are just now becoming available. Another IEEE standard called 802.22 is competing with the Wi-Fi version.

■ FIGURE 1.

Texas Instruments' SimpleLink Wi-Fi CC3000 BoosterPack kit was designed to help engineers design

new M2M and IoT applications. It speeds and simplifies Wi-Fi Internet connectivity development and uses TI's MSP430 or TivaC series of MCUs.



#10 — M2M and IoT applications

Wi-Fi is also an option for machine to machine (M2M) and the Internet of Things (IoT) uses. This is the idea of connecting devices other than computers and cell phones to the Internet. M2M applications provide remote control and monitoring functions for practically anything. M2M applications such as monitoring vending machines, storage facility temperatures, or the location of 18-wheelers use embedded cell phones and the standard cellular networks. Yet, Wi-Fi is another lower cost option where access points are available. As an example, GE, LG, and Whirlpool have enabled many of their high-end home appliances like refrigerators and washing machines with Wi-Fi so they can connect to a home network and provide feedback to the

manufacturer on appliance usage and condition.

Some estimate there will be 20 to 50 billion devices connected to the Internet in the coming years. Wi-Fi will be a part of it. Products such as Texas Instruments' new development kit in Figure 1 will make it easy.

Wi-Fi is basically everywhere these days. It is used in factory automation, chemical plants, and other industrial applications. Wi-Fi eliminates expensive wiring inside plants and has the reliability and security industry demands. Wi-Fi is also inside airplanes, and a new version of it will be used in the forthcoming Intelligent Transportation System (ITS) for roadside communications with cars and between vehicles. If you don't

Wi-Fi does provide excellent security in the form of several types of encryption. However, the original security system called wired equivalent privacy (WEP) has been shown to be too weak, so more elaborate systems were created under the IEEE 802.11i standard.

use Wi-Fi now — which is highly unlikely — you will be in the future. The good news is that Wi-Fi will continue to evolve and improve, as new versions continue to find new applications. **NV**



AP CIRCUITS
PCB Fabrication Since 1984

As low as...

\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com



The Convenient All-in-One Solution for Custom-Designed Front Panels & Enclosures



You design it
to your specifications using
our FREE CAD software,
Front Panel Designer



ONLY \$90.24
with custom
logo engraving

We machine it
and ship to you a
professionally finished product,
no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

**FRONT PANEL
EXPRESS**

FrontPanelExpress.com
1(800)FPE-9060

pattern to make transferring designs from the mini breadboard BakPak to a more permanent configuration simple. More pre-configured BakPaks are planned for the near future. The D-Axe BP main board is \$12.99 by itself, or complete with a PICAXE 20M2 for \$15.99.

For more information, contact:
Aztec MCU Prototyping
Web: www.aztecmcu.com

FUNCTION/ARBITRARY WAVEFORM GENERATOR OFFERINGS

B&K Precision has announced the launch of its 4050 series — a line of four dual channel function/arbitrary waveform generators. These new instruments can generate waveforms up to 50 MHz for use in education and applications requiring stable and precise sine, square, triangle, and pulse waveforms, with modulation and arbitrary waveform capabilities.

All models are dual channel, providing a main output voltage that can be varied from 0 to 10 Vpp into 50 ohms, and a secondary output that can be varied from 0 to 3 Vpp into 50 ohms. Their large 3.5" color LCD display, rotary control knob, and numeric keypad with dedicated waveform keys and output buttons make waveform adjustments simple.

Equipped with a 14-bit, 125 MSa/s, 16K point arbitrary waveform generator, the 4050 series provides

users 48 built-in arbitrary waveforms and the ability to create and load up to 10 custom 16 kpt waveforms using the included waveform editing software via a standard USB interface on the back. An optional USB-to-GPIB adapter is available for GPIB connectivity.

The 4050 series offers a wide variety of modulation schemes for modulated signal applications: amplitude and frequency modulation (AM/FM); double sideband amplitude modulation (DSB-AM); amplitude and frequency shift keying (ASK/FSK); phase modulation (PM); and pulse width modulation (PWM).

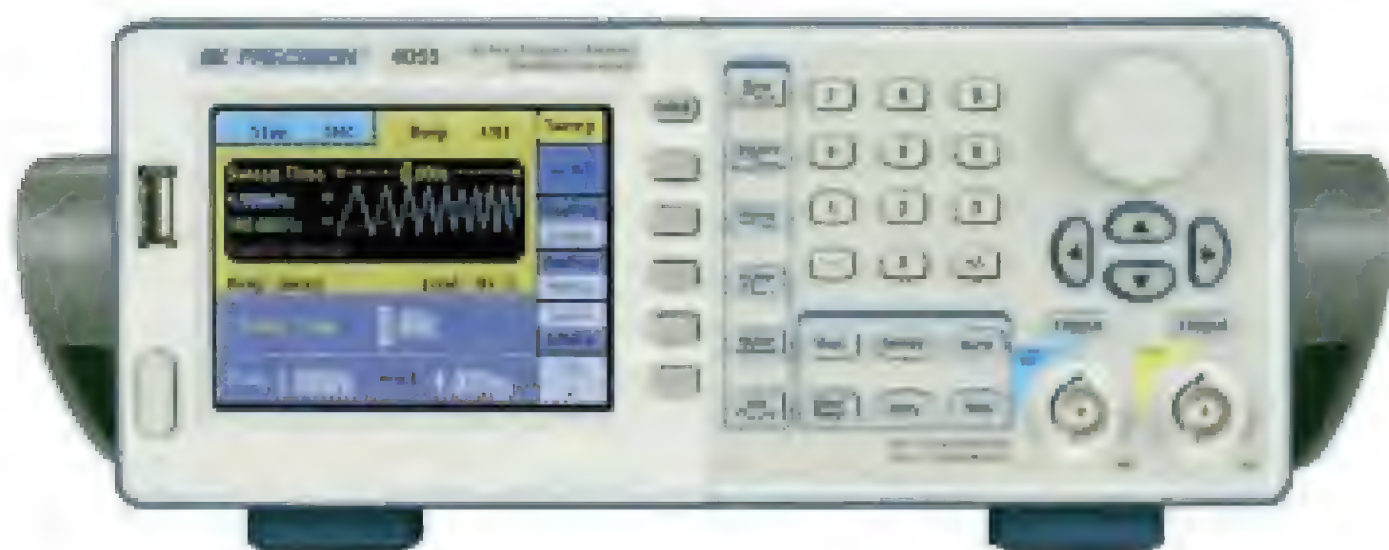
Other standard features include linear and logarithmic sweep function, built-in counter, sync output, trigger I/O terminal, and USB host port on the front panel to save and recall instrument settings and waveforms.

Additionally, a standard external 10 MHz reference clock input is provided for synchronization of the instrument to another generator — a feature not typically found in generators at this price point.

Available immediately, B&K Precision's 4050 series products are all backed by a standard three year warranty and are listed at the following prices:

- 4052 — 5 MHz — \$499
- 4053 — 10 MHz — \$599
- 4054 — 25 MHz — \$850
- 4055 — 50 MHz — \$1,050

For more information, contact:
B&K Precision
Web: www.bkprecision.com



B&K Precision is running a 10% off MSRP promotion from August 1, 2013 through November 30, 2013. Visit their website for details.

REVERSE- ENGINEERING TOOL FOR RECREATING SCHEMATICS

The RevEng System available from Saelig Company, Inc., is a reverse-engineering hardware tool for creating professional quality circuit schematic diagrams from a sample printed circuit board that lacks



documentation. The RevEng System consists of a PC-controlled continuity-detecting hardware system, RevWin control software, and EdWin — a fully featured CAD package. RevEng 'learns' the connectivity of the sample circuit, producing a netlist of the components and connections for importing into EdWin software, resulting in professional quality circuit diagrams.

RevEng's learning process is achieved through clips, connectors, and probes that are attached to clusters of components. RevWin guides the operator to place and walk the clips around the device under test. RevWin generates the most efficient sequence of clip combinations and movements to learn all possible connections, but the operator can override the automatic placement of the clips if necessary.

The RevEng System is available as an entry-level system with 256 measurement channels for low budgets or small to medium circuits. For larger, more complex circuit needs, a high pin-count system equipped with up to 2,048 measurement channels is also available. Contact Saelig for pricing.

For more information, contact:
Saelig
Web: www.saelig.com

CLASSIFIEDS

KITS/PLANS

Electronic Kits

- Fun and Educational
- Discount White and Blue LEDs
- Quantity Discounts

www.bakatronics.com

D.I.Y. LASER TAG

www.LaserTagParts.com

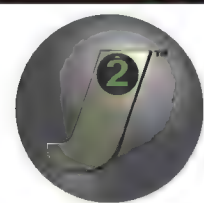
PARTS | KITS | SCHEMATICS | FIRMWARE
FORUM | STORE | HOBBY & COMMERCIAL

LIGHTING



Automatic
Stair Lighting

www.ReactiveLighting.com



- ♦LED modules
- ♦Solar controls
- ♦Dusk to dawn
- ♦Dimmers 12 V

eBay store name:

J2 LED LIGHTING LLC

<http://stores.ebay.com/j2ledlightingllc>

Subscribe today!

www.nutsvolts.com

ROBOTICS

Ultrasonic Distance Sensing Made EZ

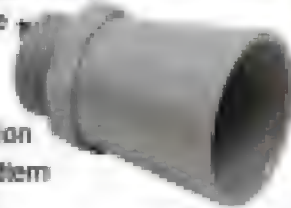
HRUSB-MaxSonar[®]-EZ™

- Multi-Sensor operation
- USB interface
- Easy integration
- 1 mm resolution
- Calibrated beam pattern
- Starting at \$49.95



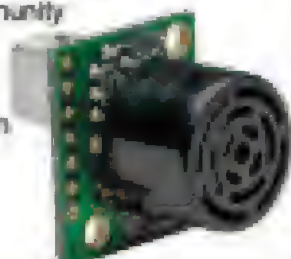
HRXL-MaxSonar[®]-WR™

- High noise tolerance
- IP67 rated
- 1 mm resolution
- Multi-Sensor operation
- Calibrated beam pattern
- Starting at \$109.95



I2CXL-MaxSonar[®]-EZ™

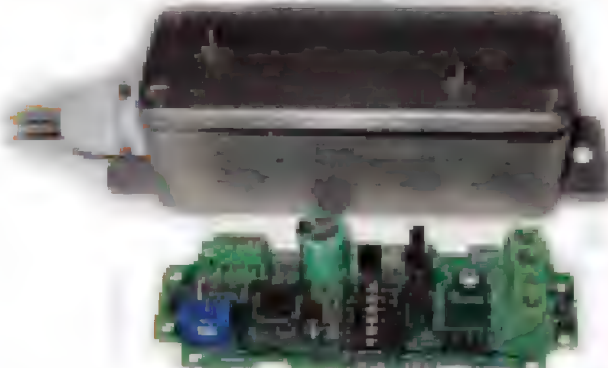
- Great for UAV's and robotics
- Incredible noise immunity
- I2C interface
- 1 cm resolution
- Automatic calibration
- Starting at \$39.95



www.maxbotix.com

QKITS.COM

Mini DC Motor Controller



Up to 8A 35V; Adjustable acceleration time and frequency. Comes assembled in small metal case.

1-888-GO 4 KITS

WANT TO BUY

WANT TO BUY: WHOLESALE OR MANUFACTURING COMPANY

Inventory, machinery, bank foreclosures, estates. ANYTHING CONSIDERED.

NO COST OR OBLIGATION!

In our 35th year! Visit

www.iseliquidator.com

for our latest list.

PHONE: 956-444-0004 888-351-5550

COMPONENTS



RF PARTS™ CO.

Tubes, Transistors, Power Components

Email: rpf@rfparts.com • Web: www.rfparts.com

800-737-2787 Fax 888-744-1943

SURPLUS

SURPLUS ELECTRONIC PARTS & ACCESSORIES



Over 20,000
Items in Stock.

Cables
Connectors
Displays
Fans

Hardware
LEDs
Motors
Potentiometers

Relays
Semiconductors
Service Manuals
Speakers

Switches
Test Equipment
Tools
VCR Parts

Surplus Material Components

SMC ELECTRONICS

www.smcelectronics.com

No Minimum Order.

Credit Cards and PAYPAL Accepted.

Flat \$4.95 per order USA Shipping.

OSCILLOSCOPES

LOW COST, VERSATILE 30MHz SCOPE



ONLY \$299

WWW.SAELIG.COM

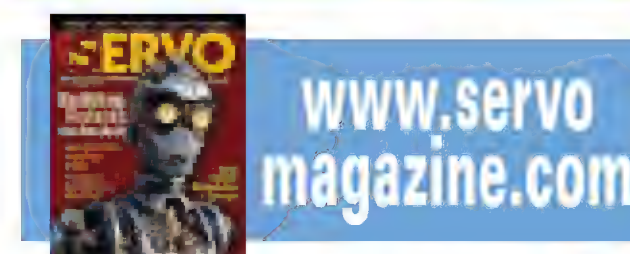
HARDWARE WANTED

DEC EQUIPMENT WANTED!!!

Digital Equipment Corp.
and compatibles.
Buy - Sell - Trade

CALL KEYWAYS 937-847-2300
or email buyer@keyways.com

ENERGY SOLUTIONS



MISC FOR SALE

HUGE CLOSE-OUT SALE!

All kinds of electronic
products and parts!

Test equipment, schematics, tubes,
audio-video equipment ... too
numerous to list. Call for availability.

loustvelectronics@verizon.net
215.271.1138 or 215.468.3028

VACUUM TUBES AND MORE



Just
Radios



Capacitors & Resistors for Tube Radios

David and Babylyn Cantelon
6 Ferncrest Gate, Scarborough
Ontario, Canada, M1W 1C2

www.justradios.com
justradios@yahoo.com
(416) 502-9128

Did You Know ...

Preferred Subscribers get access
to all the digital back issues of
Nuts & Volts for free?

Call for details
1-877-525-2539

or go to

www.nutsvolts.com/faqs

>>> QUESTIONS

Moving to SMT

I want to move to SMT components. Do I need a hot air reworking station, or just a hot air station? Can you recommend something for someone with a small budget?

#10131 **Rich Strauss**
Birmingham, AL

Arduino isn't Keeno

I developed a great project in Processing and then — when I went to port it to the Arduino — it didn't fit.

Is there a utility for Processing or some way of telling when I'm close to the hardware limits of the Arduino?

#10132 **Don Hyland**
via email

PWM Power and Short-circuit Protection

I'm trying to design a PWM circuit to control power to a resistive heater strip for my camera/telescope. Recently, my nice commercial unit let the magic smoke out after a short on the output side, so I'm not eager to replace it with another of the same make.

The problem is I'm a complete newbie. I'm using a 4093 for the PWM part of the circuit and a pMOSFET (IRF9510) for the power switching. If I didn't want the short-circuit protection, I would be done. I've found a few schematics online but am not sure how to integrate them into my circuit.

Here's the schematic (**Figure 1**) I've put together, however, there are two things that bother me. First, I've

essentially glued the short-circuit protection onto the output of the pMOSFET which means I have another diode drop in the output. That seems like it should be unnecessary and it seems like I ought to be able to put the IRF9510 where the SK100 is, but I'm not sure how to do that correctly.

Second, the whole thing will be powered off a marine battery and I'd like some input protection/isolation of the control part to avoid the possibility of frying the 4093 from transients when hooking up power.

Any help is greatly appreciated!
#10133 **Roland Roberts**
Brooklyn, NY

Can Resistors Make the Difference?

I'm working on an audiophile-quality all-transistor amp. I was told that carbon film resistors are less noisy

than wire-wound precision resistors — which are more expensive. Could I hear the difference in resistors? Is there a theoretical noise difference?

#10134 **Allan Perali**
Decatur, IL

Audio Amp ICs

I need a good, single-source audio amp IC for a project. The LM-386 isn't powerful enough. Is there a 'super' LM-386 op-amp that's also single-source (only one power connection)?

#10135 **Doug Schacter**
via email

Battery Disposal

I have a box full of old lithium batteries. I know they're not supposed to be put in the normal trash, but I don't want to pay a fee to have 'hazardous waste' removed. Is there a safe, environmental friendly way for me to dispose of the batteries that won't

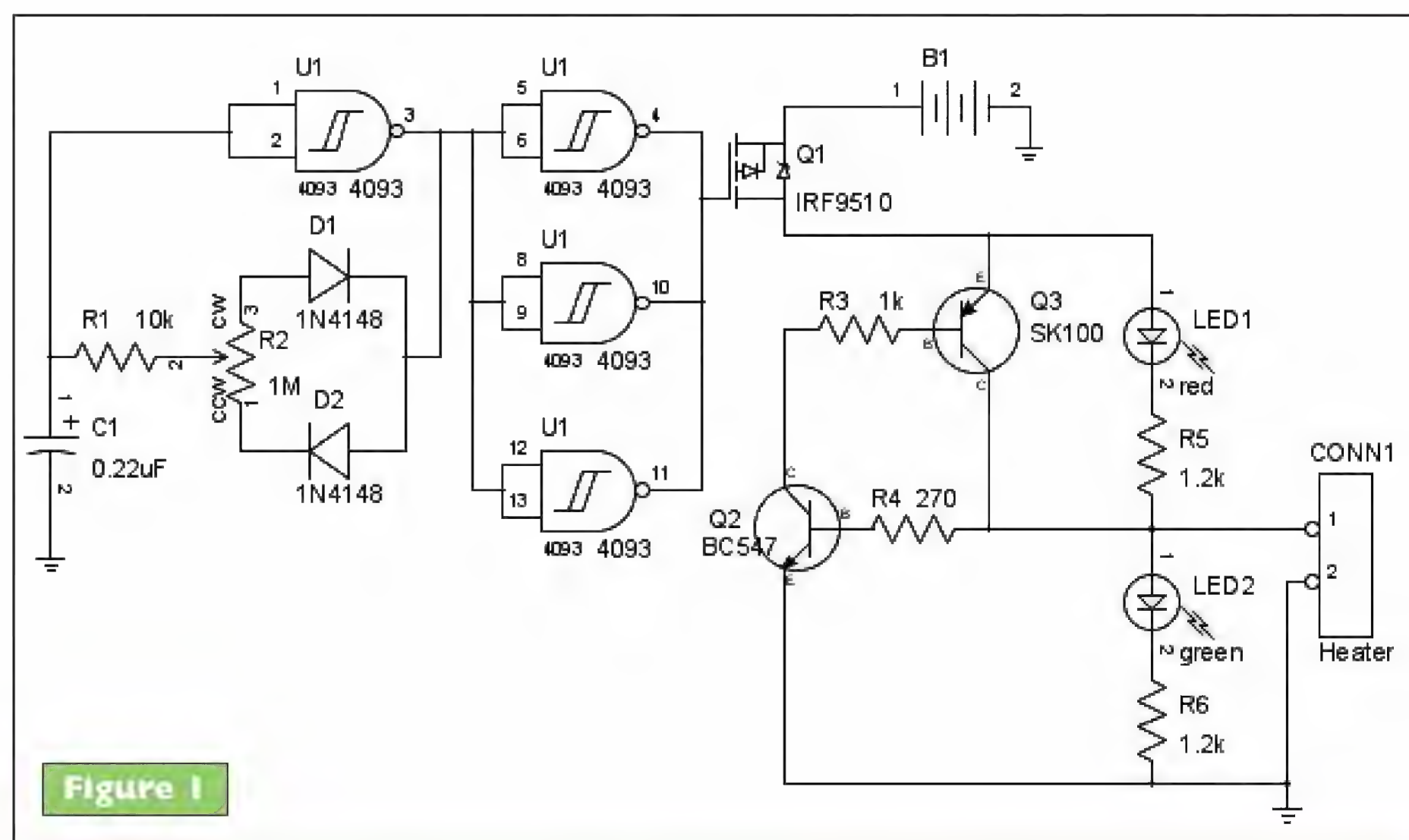


Figure 1

Send all questions and answers by email to forum@nutsvolts.com or via the online form at www.nutsvolts.com/tech-forum

All questions AND answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. All

submissions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted by readers and **NO GUARANTEES**

WHATSOEVER are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgment!

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

cost me time and money?

**#10136 Erik Prichard
Ft Wayne, IN**

Where Does a Kid Start?

My 10 year old wants to start building circuits. What's a good resource for easy, safe, one hour projects for the younger set?

**#10137 Judy Boyer
via email**

Surplus Parts for Amps

I'm just getting into guitar amplifiers, and I'm having a hard time finding parts — tubes, transformers, and high-voltage capacitors that are affordable. What are the best surplus sources, short of taking apart old amps?

**#10138 Steven Sewal
via email**

Convert USB Webcam into Analog Video Source

Used USB webcams are a dollar a dozen these days — much cheaper than native analog-out cameras (i.e., "security" cams). Is it possible or feasible to convert the USB cams to analog out? I presume it depends on the video chip used in the webcam — whether it has USB integrated or if that's a function of the small PCB on which it's attached. Any pointers, tips, advice, etc., would be greatly appreciated.

**#10139 Dave T.
via email**

Analog vs. Digital Meter

I have an old Triplet analog multimeter and a new, no-name digital multimeter. A friend told me that, even though I can read out voltage to the second decimal place, my old Triplet is more accurate. I'm confused about the accuracy/precision difference. Can you clarify?

**#101310 Katy Aricanli
Phoenix, AZ**

Save a Light

I'm constantly replacing flashlights because of swollen D cells that expand and get stuck in the flashlight barrel. Is there some way to remove the cells without ruining the flashlight or splattering battery gunk all over?

**#101311 Mitchel Westervelt
Denver, CO**

>>> ANSWERS

[#8135 - August 2013] Circuit Simulation Program

I'm looking for a circuit simulation program that is inexpensive and easy for a novice to learn. I'm mostly interested in analog circuits, but I do dabble with digital from time to time. I want to be able to change component values and analyze circuit behavior. This is for my own enjoyment and education, not for professional use. Can someone recommend a program they are using or have prior experience with?

Here's a nice free circuit simulator:
www.falstad.com/circuit/.

Linear Technology makes a nice free SPICE program for when you graduate to more advanced design methodologies (www.linear.com/designtools/software/#LTspice).

To further your education, Digilent has some nice online instruction for analog at:

[www.digilentinc.com/
Classroom/RealAnalog/](http://www.digilentinc.com/Classroom/RealAnalog/)

[www.digilentinc.com/Products/
Detail.cfm?NavPath=2,842,
1018&Prod=ANALOG-DISCOVERY](http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,842,1018&Prod=ANALOG-DISCOVERY)
Or

[www.analog.com/en/
university/topic.html](http://www.analog.com/en/university/topic.html).

For digital, try:

[www.digilentinc.com/
classroom/realdigital/](http://www.digilentinc.com/classroom/realdigital/)

[www.digilentinc.com/classroom/
Electronics101/index.cfm](http://www.digilentinc.com/classroom/Electronics101/index.cfm).

As a side note, as to textbooks, I'm impressed by:

[www.ntspress.com/publications/
circuits-second-edition/](http://www.ntspress.com/publications/circuits-second-edition/)

Unfortunately, it's a bit spendy.

Here's a site on basic car audio electronics which is very engaging: www.bcae1.com/. This one is geared towards the car audio enthusiast who wants to learn about and repair his equipment:

[www.bcae1.com/repairbasics
forbcae1/repairbasics.htm](http://www.bcae1.com/repairbasicsforbcae1/repairbasics.htm).

**Charles Petras
via email**

[#9132 - September 2013] Arduino

I've hit the performance wall with my Arduino processor. I'm using the standard development environment and the latest version of the cards from SparkFun. Where do I go from here?

#1 There are a few options for higher performance processors that are moderately compatible with basic Arduinos. My personal favorite is the Maple from **Leaflabs.com**. Some others are the Arduino Due, and the chipKIT boards from Digilent. Unfortunately, all of these boards run at 3.3V which often means changes to external hardware are needed. The Maple and chipKIT boards have several 5V input compatible digital pins, and 5V analog input is possible using two external resistors.

All three boards will require some modification to anything but the most basic code. The Maple will require the most changes, but also has the best access to hardware features (timers, high speed ADC, etc.) due to those changes. The Due will require the fewest changes, and has the best support for standard libraries. The Maple and Due have the best documentation. If you like the look of these boards, make sure to check that

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

libraries you use will be supported.

If 5V is a must, you may have to consider switching development platforms. Both Atmel AVR and Microchip PIC microcontrollers are available that run at much higher speeds and have better peripheral sets while still using 5V. I'm afraid I don't know as much about the AVR line, but PIC18F, 24F, and 30F are all quite powerful controllers. If you decide to go this route, be warned that the learning curve is very steep at the start. Don't forget to compare performance by MIPS (million instructions per second), not clock speed.

Another 5V solution is to use a second Arduino. Using two board lets you do twice the work at the same speed. You can communicate between them using serial or SPI/I²C with the wire library. This can be a lot of work, but it also may be the simplest solution to some problems. You may want to simply see if you can improve your coding abilities.

Most people underestimate what can be accomplished with a basic Arduino or microcontroller. For instance, if your code has delays of more than 1 ms, there is probably a faster way to run the same code. Unfortunately, the Arduino libraries are not really designed for the highest performance. For example, the only interrupts that are available using the core libraries are for pin changes. Many common programming techniques to improve performance rely on timer, serial, and ADC interrupts. If your code isn't proprietary, don't be afraid to post it to the Arduino forums to ask if there is a faster way to do the same thing. If you already use the best practices, they will also be able to point you toward an upgrade that fits most closely with what you need to accomplish.

Sam
Jersey Shore, PA

#2 The Arduino is a great learning platform, but it does have limitations.

If the issue is math crunching ability, then you might look into a book on algorithms to make certain you haven't missed any computational tricks. Another avenue is to use Assembler. I've written directly to the I/O hardware to achieve a nearly 100x improvement in I/O handling. Lastly, consider upgrading to a new card, such as the chipKit series of Arduino compatible cards. The problem with this approach is cost (about the same as a stock Arduino) and you'll have to use the proprietary chipKit development environment. As a result, you probably won't be able to port your code back to a generic Arduino.

Les Larkin
via email

[#9133 - September 2013] Need to See IR Light

I'm working with IR LEDs for a wireless stereo system in my living room. I'm having trouble visualizing the dead spots (shadows). Do you know any way to see the IR light distribution, short of me buying a pair of those night goggles from the military?

Most digital cameras are infrared (IR) sensitive. The simple test is to point an IR remote control at the camera and watch the LCD screen. For the "dead spots" or shadow tests, turn all visible lights off, then turn on your IR source. Look around the living room with the digital camera (I recommend lowering the screen brightness), and see where there is no illumination with the IR source.

Raymond Ramirez
Bayamon, PR

>>> ANOTHER RESPONSE

[#7132 - July 2013] Tech Jobs

I read your question and response on Tech Jobs in the August 2013 issue. Here is another response:

Your son should consider the local community college, and look at other careers related to electronics. While engineers are often mentioned, technicians with associate's degrees are now more in demand (USA TODAY).

Electronics is just one option — also consider other related fields such as machine tool technology, industrial electronics, instrumentation repair, or machine maintenance.

These programs include electronics along with motor control, programmable logic controllers, instrumentation, fluid power, sensors, pumps, valves, process control, and industrial robotics.

These people are in demand and command high incomes.

For info on career options and certifications, check out the Electronics Technicians Association International at www.eta-i.org.

Glen W. Spielbauer
Dallas, TX

[#7131 - July 2013] Voltage Filter

One of the suggestions was one that I was first tempted to suggest, but did not as it is *potentially* dangerous. The use of a filtering capacitor to dampen the voltage should not be tried. The sender circuit is low impedance and needs a very large value capacitor to be successful. Unfortunately, that capacitor will store quite a bit of energy and the sender rheostat — in the presence of fuel and air — could create a spark that is sufficient to explosively ignite the fuel vapor. (Think TWA-800.)

You have gotten a lot of good suggestions. The best ones deal with repairing the system as it was designed. You might want to make sure the sender is grounded properly and try an external wire back to the gauge to bypass the original one as a test.

Joe Leikhim
Oviedo, FL

AMATEUR RADIO AND TV

Ramsey Electronics82-83

BATTERIES/CHARGERS

Cunard Associates27

BUYING ELECTRONIC

SURPLUS

Jaycar Electronics.....43

Weirdstuff Warehouse27

CCD CAMERAS/VIDEO

Ramsey Electronics82-83

CIRCUIT BOARDS

AP Circuits75

Cunard Associates27

Digilent5

Dimension Engineering.....6

ExpressPCB18

Front Panel Express LLC75

Futurlec59

PCB Pool58

CNC

Dynomotion59

COMPONENTS

Cana Kit Corp.67

Futurlec59

COMPUTER

Hardware

Weirdstuff Warehouse27

Microcontrollers / I/O Boards

Abacom Technologies58

Digital 6 Laboratories.....27

Dynomotion59

microEngineering Labs.....42

MikroElektronika3

Pololu Robotics & Electronics...19

Software

Labcenter Electronics .Back Cover

DESIGN/ENGINEERING/

REPAIR SERVICES

Cana Kit Corp.67

ExpressPCB18

Front Panel Express LLC75

Labcenter Electronics .Back Cover

PCB Pool58

EDUCATION

Command Productions41

Digilent5

NKC Electronics27

Poscope.....52

EMBEDDED TOOLS

NetBurner2

ENCLOSURES

Front Panel Express LLC75

KITS & PLANS

Cana Kit Corp.67

Jaycar Electronics.....43

NetBurner2

NKC Electronics27

QKITS27

Ramsey Electronics82-83

MISC./SURPLUS

All Electronics Corp.47

Front Panel Express LLC75

Weirdstuff Warehouse27

MOBILE RADIO MOUNTS

E Tip Inc.27

PROGRAMMERS

Futurlec59

microEngineering Labs42

MikroElektronika3

RF TRANSMITTERS/

RECEIVERS

Abacom Technologies58

Digital 6 Laboratories.....27

E Tip Inc.27

ROBOTICS

Digilent5

Dynomotion59

Lemos International Co.27

Pololu Robotics & Electronics...19

SATELLITE

Lemos International Co.27

SECURITY

E Tip Inc.27

SOLDERING STATIONS

AND IRONS

HAKKO25

TEST EQUIPMENT

Dimension Engineering.....6

Jaycar Electronics.....43

NKC Electronics27

Poscope.....52

TOOLS

HAKKO25

MikroElektronika3

NetBurner2

PanaVise35

Poscope.....52

WIRELESS PRODUCTS

Digital 6 Laboratories.....27

Abacom Technologies58

All Electronics Corp.47

AP Circuits75

Cana Kit Corp.67

Command Productions41

Cunard Associates27

Digilent5

Digital 6 Laboratories.....27

Dimension Engineering.....6

Dynomotion59

E Tip Inc.27

ExpressPCB18

Front Panel Express LLC 75

Futurlec59

HAKKO25

Jaycar Electronics43

Labcenter

ElectronicsBack Cover

Lemos International Co.27

microEngineering Labs42

MikroElektronika3

NetBurner2

NKC Electronics27

PanaVise35

PCB Pool58

Pololu Robotics &

Electronics19

Poscope.....52

QKITS27

Ramsey Electronics82-83

Weirdstuff Warehouse27



RAMSEY

Spooktacular Halloween Treats!



Blinky-Eyes Animated Display

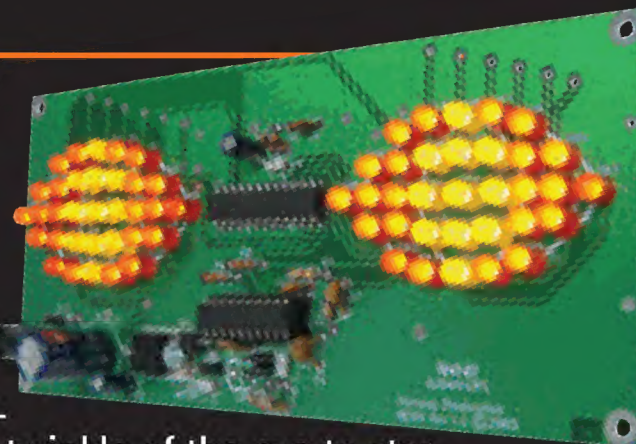
- ✓ Animated display of 66 super bright LED's!
- ✓ Microcontroller controlled!
- ✓ Changes brightness automatically!
- ✓ Animated with constant motion!

The ultimate animated LED kit that will dazzle you and delight your friends! Uses a microcontroller to randomly select from many different animations such as a long pause before a wink, or a twinkle of the eye to startle passers-by!

Four modes to satisfy any enthusiast's desires: 1. Off for long random periods, then blinks or winks. Designed to scare! 2. On for long periods before performing an animation, perfect for costumes and displays! 3. Animates all the time for constant motion, perfect display attention-getter. 4. Random fire! When placed in a pumpkin will light it up like you wouldn't believe! As if this weren't enough, the BE66 can also control a small hobby motor to shake bushes at random intervals or signal an external player to make a scary sound! Also has a CDS cell to sense light. In one mode, the display will dim as it gets dark for battery operation, and in the other mode it will turn off when it's too bright, so it plays only in the dark!

BE66 Blinky-Eyes Animated Display Kit

\$59.95



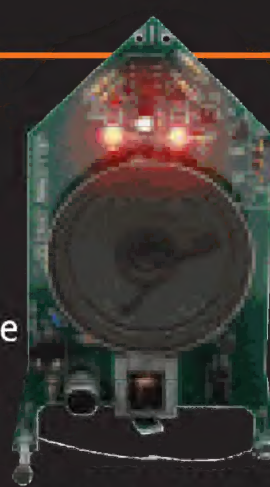
Automatic Animated Ghost

- ✓ Automatically greets your visitors!
- ✓ Responds to sudden noises!
- ✓ Built-in microphone!
- ✓ Adjustable sensitivity

Who says ghosts are make believe? Once your friends come upon this one they'll think differently! The unique circuit board design includes two ominous blinking eyes that change with various conditions, including sudden changes in ambient noise. A highly sensitive built-in microphone picks up anything from noises to talking and makes the ghost dance with its built-in motor, make eerie sounds with the built-in speaker, and randomly blink. A white cloth and a hanger are included as shown to make it look like the real thing. Runs on 2 AAA batteries (Not included).

MK166 Automatic Animated Ghost Kit

\$21.95



Laser Light Show

- ✓ Audio input modulates pattern!
- ✓ Adjustable pattern & size!
- ✓ Projects neat motorized patterns!
- ✓ Uses safe plastic mirrors!

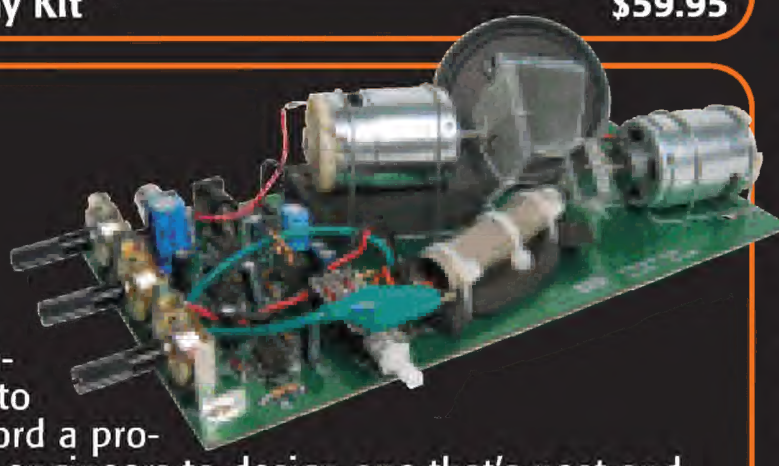
You've probably seen a laser show at concerts or on TV. They're pretty impressive to say the least! Knowing that you can't afford a professional laser display we challenged our engineers to design one that's neat and easy to build, yet inexpensive.

Well, the result is the LLS1 Laser Light Show! This thing is sweet and perfect for your haunted house or halloween parties! It utilizes two small motors and a small standard laser pointer as the basics. Then, we gave it variable pattern and speed controls to customize the pattern!

Not enough, you say? How about a line level audio input to modulate the pattern with your CD's, music, or spooky sound effects? You bet! Everything is included, even the small laser pointer. Runs on 6-12 VDC or our standard AC adapter.

LLS1 Laser Light Show Kit
AC121 12VDC 1A Regulated Power Supply

\$49.95
\$9.95



Halloween Pumpkin

- ✓ 25 bright LED's!
- ✓ Random flash simulates flickering candle!
- ✓ Super bright LED illuminates entire pumpkin!
- ✓ Simple & safe 9V battery operation

The perfect "starter" kit with a terrific Halloween theme! You won't be scraping the seeds and guts out of this pumpkin! Six transistor circuit provides a neat random flash pattern that looks just like a flickering candle. Then a super bright LED illuminates the entire pumpkin with a spooky glow!

The pumpkin face is the actual PC board, and assembly is easy through-hole soldering of all components and LED's. Your pumpkin is powered by a standard 9V battery (not included) which snaps to the back of the pumpkin. An on/off switch is also included. Create a new kind of pumpkin this year, and learn about LED's and electronics at the same time!

MK145 Electronic Halloween Pumpkin Kit

\$11.95



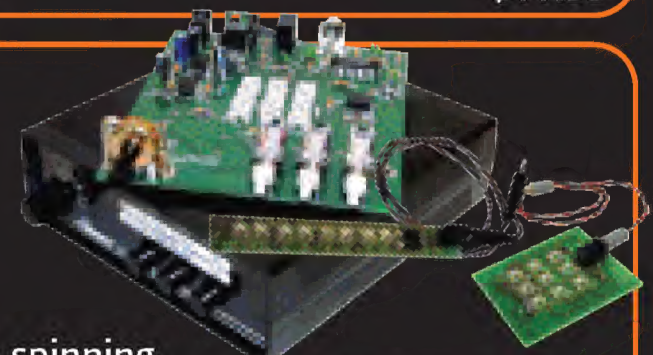
High Power LED Strobe Light

- ✓ Everlasting LEDs won't burn out!
- ✓ Variable flash rate and audio trigger!
- ✓ Bass and treble trigger modes
- ✓ Safe low voltage construction, no fragile high voltage Xenon tube to break!

Everyone has seen Xenon stroboscopes "stop" a spinning quarter or flash to music. Now you can get the same effect without the expense and hassle of high voltage Xenon tubes! Great for Halloween displays and parties!

A plug-in 3x3 array of super bright Telux™ LEDs creates a brilliant, sharp flash similar to a Xenon flash tube. The LEDs can also be mounted directly on the main PC board or on a remote display board. Optional display boards also available. Variable flash rate from 1-220 FPS plus an audio flash trigger with selectable bass or treble triggering. Easy to connect with standard stereo RCA audio connectors. External trigger can be looped to additional units for simultaneous flash displays from the same source! Runs on safe 12-15VDC so a great kit for the kids to build!

LEDS1C	High Power LED Strobe Light Kit with Case	\$49.95
LEDS8	Display Board, Inline Array of 8 LEDs	\$17.95
LEDS20	Display Board, 5x4 Array of 20 LEDs	\$29.95
AC121	12VDC 1A Regulated Power Supply	\$9.95



Tri-Field Meter & "Ghost Detector"

- ✓ See electric, magnetic, and RF fields!
- ✓ Watch the magnetic fields of the earth!
- ✓ Sense different magnetic poles
- ✓ Detect RF transmitter fields
- ✓ Graphical LED display allows you to "see" the invisible fields
- ✓ Great learning tool for EMF, RF, and magnetic field theory

THE GHOST DETECTOR!

Call it a Tri-Field Meter, an Electrical, Magnetic, and RF Detector, a Ghost Detector, or a Tricorder, but walking around with this on Halloween will seriously impress even the most doubting of Thomases out there!

The TFM3C has three separate field sensors that are user selectable to provide a really cool readout on two highly graphical LED bargraphs! Utilizing the latest technology, including Hall Effect sensors, you can walk around your house and actually "SEE" these fields around you! You will be amazed at what you see. How sensitive is it? Well, you can see the magnetic field of the earth... THAT'S sensitive!

The technical applications are endless. Use it to detect radiation from monitors and TV's, electrical discharges from appliances, RF emissions from unknown or hidden transmitters and RF sources, and a whole lot more! If you're wondering whether your wireless project or even your cell phone is working, you can easily check for RF! A 3-position switch in the center allows you to select electric, magnetic, or RF fields. A front panel "zero adjust" allows you to set the sensors and displays to a known clean "starting point." If the TFM3C looks familiar, it's probably because you saw it in use on the CBS show Ghost Whisperer! It was used throughout one episode (#78, 02-27-2009) to detect the presence of ghosts!

The concept is simple, it is believed (by the believers!) that ghosts give off an electric field that can be detected with the appropriate equipment. In the electric mode, the TFM3C's displays will wander away from zero even though there isn't a clear reason for it (not scientifically explainable, aka paranormal!). This would mean something has begun to give off an electric field. What it was in the Ghost Whisperer was a friendly ghost. What it will be in your house... who knows! Makes a GREAT leaning project besides! Requires 4 AA batteries.

TFM3C Tri-Field Meter Kit With Case

\$74.95



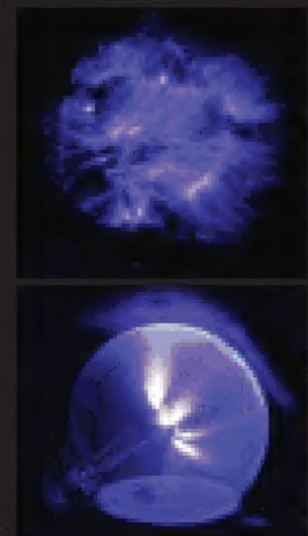
Spark Generating HV Plasma Generator

- ✓ Generate 2" sparks to a handheld screwdriver!
- ✓ Light fluorescent tubes without wires!
- ✓ Build your own Plasma Balls!
- ✓ Generates up to 25kV @ 20 kHz from a solid state circuit!

This popular kit was conceived by one of our engineers who likes to play with things that can generate large, loud sparks, and other frightening devices! And at Halloween there's no better effect than high voltage sparks flying through the air! The PG13 Plasma Generator creates a very impressive 25,000 volts at 20 kHz, to provide a stunning display of high voltage! It will draw a cool looking 2" spark to a hand held screwdriver, or light fluorescent tubes without any connection!

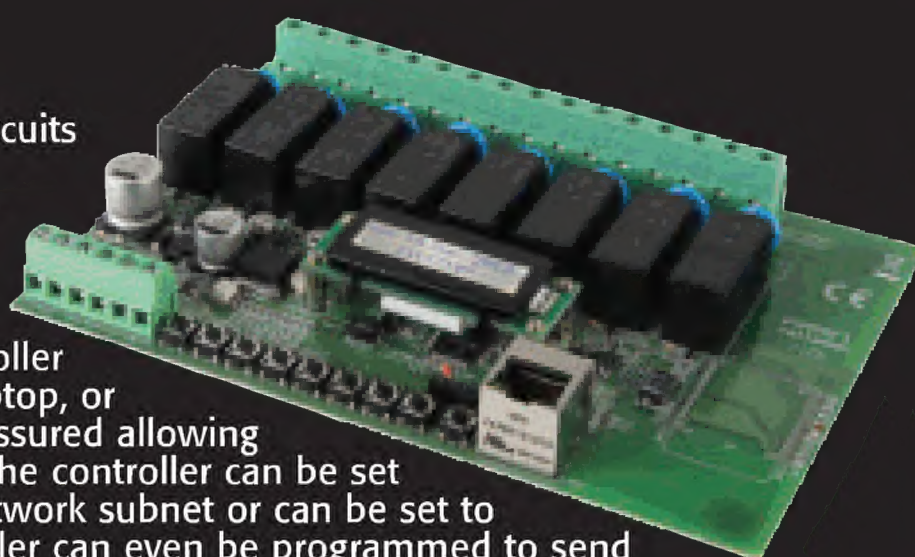
It produces stunning lighting displays, drawing big sparks, to perform lots of high voltage experiments. In the picture, we took a regular clear "Decora" style light bulb and connected it to the PG13 - WOW! A storm of sparks, light tracers and plasma filled the bulb. Holding your hand on the bulb doesn't hurt a bit and you can control the discharge! It can also be used for powering other experiments; let your imagination be your guide! Operates on 16VAC/VDC for maximum output. Can also be run from 5-16VAC/VDC to reduce the output voltage.

PG13	Plasma Generator Kit	\$64.95
PS21	110VAC Input, 16VAC Output, Power Supply	\$19.95



8-Channel Ethernet Remote Controller

Now you can easily control and monitor up to 8 separate circuits via the standard Ethernet network in your home or office. Connection wise it couldn't be simpler. The controller functions as an IP based web server, so it can be controlled by any internet browser that can reach your network! There are no drivers or proprietary software required, just access the controller like any web page from your PC, laptop, or even your smartphone! Security is assured allowing up to 4 separate user credentials. The controller can be set to a specific static IP within your network subnet or can be set to DHCP (auto negotiate). The controller can even be programmed to send you an email to notify and confirm power up and status changes!



To simplify the connection of your equipment to the controller, 8 separate and isolated relay outputs are provided! This gives you internet or network control of up to 8 separate functions. No need to worry about interfacing a logic high or logic low, or burning up the interface! The applications are endless! From something as simple as turning on and monitoring lights at your house with a normal latched closure to advanced control of your electronic gadgets, radio equipment, or even your garage door! Each relay contact is rated at 12A at 30VDC or 16A at 230VAC.

Each of the 8 channels has built-in timer and scheduler programs for day, weekend, working days, every day, and every day except Sunday. Relay control functions are programmable for on, off, or pulse (1-99 seconds, 1-99 minutes, or 1-99 hours). In addition to control functions, the web interface also displays and confirms the status of each channel. Each channel can be custom labeled to your specific function name. The controller operates on 12VDC or 12VAC at 500mA or our new AC121 global 12VDC switching power supply over to the left! Factory assembled, tested, and ready to go! Even includes a Cat-5 cable!

VM201 8-Channel Remote Ethernet Controller, Factory Assembled & Tested \$169.95

Electronic Police Siren

Exactly duplicates the upward and downward wail of a police siren. Switch closure produces upward wail, releasing it makes it return downward. Produces a loud 5W output, and will drive any speaker! Horn speakers sound the best! Runs on 6-12VDC.



SM3 Electronic Siren Kit \$7.95

LED Windmill Display

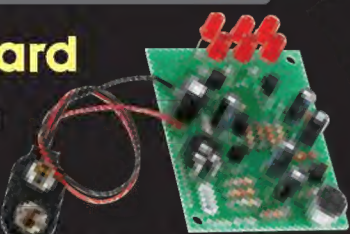
This "Windmill" displays 31 dazzling bright red LEDs in a rotating animated pattern that is sure to grab attention! The rotational speed is adjustable with a built-in pot. Runs on 9-12VDC.



CK176 LED Windmill Display Kit \$24.95

LED Sound-To-Light Board

What a neat little kit! It starts with an on-board ultra sensitive mic that captures ambient, spoken, and music audio. It then converts it to a neat 6-LED display. Great for hearing impaired alarms! Adjustable sensitivity. Runs on a 9V battery.



CK457 LED Light To Sound Display Kit \$15.95

Laser Trip Sensor Alarm

Just like the big concerts, you can impress your friends with your own laser light show! Audio input modulates the laser display to your favorite music! Adjustable pattern & speed. Runs on 6-12VDC.



LTS1 Laser Trip Sensor Alarm Kit \$29.95

Mad Blaster Warble Alarm

If you need to simply get attention, the "Mad Blaster" is the answer, producing a LOUD ear shattering raucous racket! Super for car and home alarms as well. Drives any speaker. Runs on 9-12VDC.



MB1 Mad Blaster Warble Alarm Kit \$9.95

20 Watt Mini Audio Amp

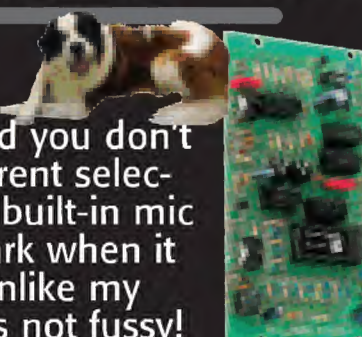
Delivers a super clean 20W output from one SMT package! Ultra efficient class D design produces no heat. PCB can be snapped into a small circle for special applications. Runs on 18VDC for rated output, or down to 10VDC for reduced output.



UAM2 20W Subminiature Amp Kit \$52.95

Electronic Watch Dog

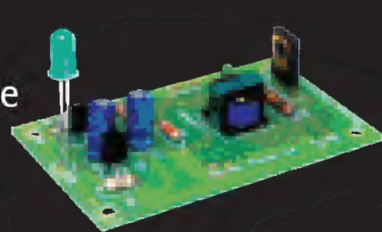
A barking dog on a PC board! And you don't have to feed it! Generates 2 different selectable barking dog sounds. Plus a built-in mic senses noise and can be set to bark when it hears it! Adjustable sensitivity! Unlike my Saint, eats 2-8VAC or 9-12VDC, it's not fussy!



K2655 Electronic Watch Dog Kit \$39.95

Tickle-Stick Shocker

The kit has a pulsing 80 volt tickle output and a mischievous blinking LED. And who can resist a blinking light and an unlabeled switch! Great fun for your desk, "Hey, I told you not to touch!" Runs on 3-6 VDC.



TS4 Tickle Stick Kit \$9.95

Van de Graaff Generators

Create your own lightning with these time tested student favorites! Produces low current static currents that can be "shocking" but perfectly safe! Draw sparks to a screwdriver, grab hold and watch your hair stand straight up! Two models produce from 200kV to 400kV!



VG Van de Graaff Generators from \$139.95

Passive Aircraft Monitor

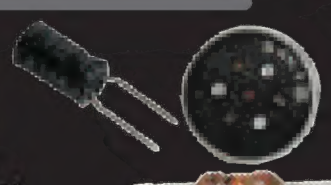
The hit of the decade! Our patented receiver hears the entire aircraft band without any tuning! Passive design has no LO, therefore can be used on board aircraft! Perfect for air shows, hears the active traffic as it happens! Available kit or factory assembled.



ABM1 Passive Aircraft Receiver Kit \$89.95

Electret Condenser Mic

This extremely sensitive 3/8" mic has a built-in FET preamplifier! It's a great replacement mic, or a perfect answer to add a mic to your project. Powered by 3-15VDC, and we even include coupling cap and a current limiting resistor! Extremely popular!



MC1 Mini Electret Condenser Mic Kit \$3.95

12VDC Regulated Supply

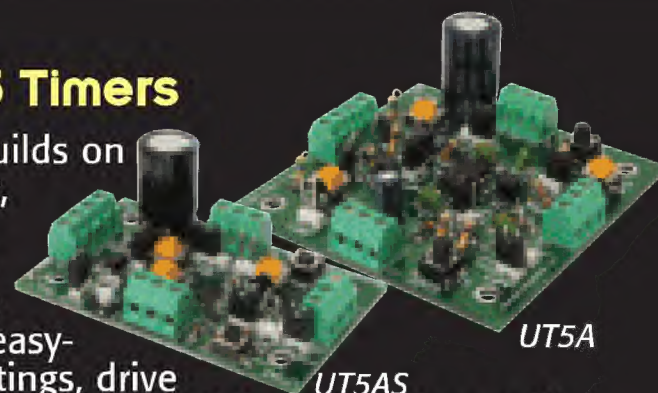
Go green with our new 12VDC 1A regulated supply. Worldwide input 100-240VAC with a Level-V efficiency! It gets even better, includes DUAL ferrite cores for RF and EMI suppression. All this at a 10 buck old wallwart



AC121 12VDC 1A Regulated Supply \$9.95

Ultimate 555 Timers

This new series builds on the classic UT5 kit, but takes it to a whole new level! You can configure it on the fly with easy-to-use jumper settings, drive relays, and directly interface all timer functions with onboard controls or external signals.

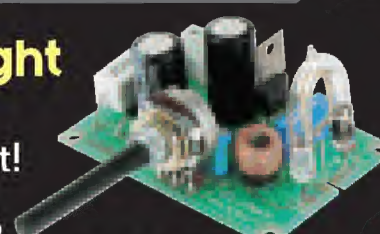


All connections are easily made through terminal blocks. Plus, we've replaced the ceramic capacitor of other timer kits with a Mylar capacitor which keeps your timings stable over a much wider range of voltages! Available in through hole or surface mount versions! Visit www.ramseykits.com for version details.

UT5A Through Hole 555 Timer/Osc Kit \$21.95
UT5AS SMT 555 Timer/Osc Kit \$29.95

Xenon Tube Strobe Light

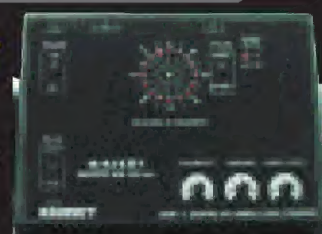
Create amazing effects with an authentic Xenon tube strobe light! Creates a super bright white FLASH with a variable speed of 2 to 20 flashes per second. Just connect 110VAC and you have a complete strobe light!



K5300 Xenon Tube Strobe Light Kit \$19.95

Doppler Direction Finder

Track down jammers and hidden transmitters with ease! 22.5 degree bearing indicator with adjustable damping, phase inversion, scan and more. Includes 5 piece antenna kit. Runs on 12VDC vehicle or battery power.



DDF1 Doppler Direction Finder Kit \$169.95

USB PIC Programmer

Finally, a compact USB PIC Programmer with a 20 pin ZIF socket for easy programming of most Microchip PIC Flash devices that does not require low voltage programming. Plus it uses USB therefore no more RS232 compatibility blues!



CK1301 USB PIC Programmer Kit \$34.95

Audio Recorder/Player

Record and playback up to 8 minutes of messages from this little board! Built-in condenser mic plus line input, line & speaker outputs. Adjustable sample rate for recording quality. 4-switch operation that can be remote controlled! Runs on 9-12VDC at 500mA.



K8094 Audio Recorder/Player Kit \$32.95

Classic Nixie Tube Clocks

Our next-gen Nixie clock kits for 2013! Hand made teak-maple, polished aluminum, or clear plexi base with options including GPS timebase! Check our site!



NIXIE Classic Nixie Clock Kits from \$229.95

Broadband RF Preamp

Need to "perk-up" your counter or other equipment to read weak signals? This preamp has low noise and yet provides 25dB gain from 1MHz to well over 1GHz. Output can reach 100mW! Runs on 12 volts AC or DC or the included 110VAC PS. Assmb.



PR2 Broadband RF Preamp \$69.95

12VDC Worldwide Supply

It gets even better than our AC121 to the left! Now, take the regulated Level-V green supply, bump the current up to 1.25A, and include multiple blades for global country compatibility! Dual ferrite cores!



PS29 12VDC 1.25A Worldwide Supply \$19.95

GET THE NUTS AND VOLTS DISCOUNT!

Mention or enter the coupon code **NVRMZ12** and receive 10% off your order!

Prices, availability, and specifications are subject to change. Not responsible for typos, stupid, printer's bleed, or really weird Halloween tricks! Visit www.ramseykits.com for the latest pricing, specials, terms and conditions. Thanks Robin... for reminding me of this ad deadline and that Halloween is approaching! Therefore today's color theme is Pantone 021C Orange! Copyright 2013 Ramsey Electronics®...so there!

800-446-2295
www.ramseykits.com

RAMSEY ELECTRONICS®

590 Fishers Station Drive
Victor, NY 14564
(800) 446-2295
(585) 924-4560

CAD CONNECTED



Proteus Version 8
Launch Offer - SAVE 10%
Use NVL2013ATJ
When placing your order

PROTEUS DESIGN SUITE VERSION 8

Featuring a brand new application framework, common parts database, live netlist and 3D visualisation, a built in debugging environment and a WYSIWYG Bill of Materials module, Proteus 8 is our most integrated and easy to use design system ever. Other features include:

- Hardware Accelerated Performance.
- Unique Thru-View™ Board Transparency.
- Over 35k Schematic & PCB library parts.
- Integrated Shape Based Auto-router.
- Flexible Design Rule Management.
- Polygonal and Split Power Plane Support.
- Board Autoplacement & Gateswap Optimiser.
- Direct CAD/CAM, ODB++, IDF & PDF Output.
- Integrated 3D Viewer with 3DS and DXF export.
- Mixed Mode SPICE Simulation Engine.
- Co-Simulation of PIC, AVR, 8051 and ARM MCUs.
- Direct Technical Support at no additional cost.

labcenter  www.labcenter.com
Electronics

**Visit our website or
phone 866.499.8184
for more details**

Labcenter Electronics North America 411 Queen St, Newmarket, Ont. Canada L3Y 2G9
Email: info@labcenter-electronics.com Tel: 905.898.0665 Fax 905.898.0683